

Karin Satie Komati

**UMA ABORDAGEM NÃO-SUPERVISIONADA PARA
SEGMENTAÇÃO DE CENAS NATURAIS COLORIDAS**

Tese apresentada ao Programa de Pós-Graduação em Engenharia Elétrica do Centro Tecnológico da Universidade Federal do Espírito Santo, como requisito parcial para obtenção do Grau de Doutor em Engenharia Elétrica.

Orientador: Prof. Dr. Evandro Ottoni Teatini Salles.

Orientador: Prof. Dr. Mário Sarcinelli-Filho.

VITÓRIA
2011

Dados Internacionais de Catalogação-na-publicação (CIP)
(Biblioteca Central da Universidade Federal do Espírito Santo, ES, Brasil)

- K81a Komati, Karin Satie, 1971-
Uma abordagem não-supervisionada para segmentação de
cenais naturais coloridas / Karin Satie Komati. – 2011.
162 f. : il.
- Orientador: Evandro Ottoni Teatini Salles.
Coorientador: Mário Sarcinelli-Filho.
Tese (Doutorado em Engenharia Elétrica) – Universidade
Federal do Espírito Santo, Centro Tecnológico.
1. Processamento de imagens. 2. Sistemas de reconhecimento
de padrões. 3. Visão por computador. 4. Fractais. 5. JSEG. I.
Salles, Evandro Ottoni Teatini. II.Sarcinelli Filho, Mário. III.
Universidade Federal do Espírito Santo. Centro Tecnológico. IV.
Título.

CDU: 621.3

Karin Satie Komati

**UMA ABORDAGEM NÃO-SUPERVISIONADA PARA
SEGMENTAÇÃO DE CENAS NATURAIS COLORIDAS**

Tese submetida ao programa de Pós-Graduação em Engenharia Elétrica do Centro Tecnológico da Universidade Federal do Espírito Santo, como requisito parcial para a obtenção do Grau de Doutor em Engenharia Elétrica.

Aprovada em 16 de dezembro de 2011.

COMISSÃO EXAMINADORA



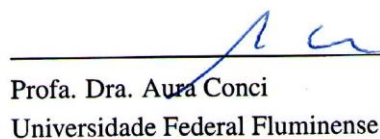
Prof. Dr. Evandro Ottoni Teatini Salles
Universidade Federal do Espírito Santo
Orientador



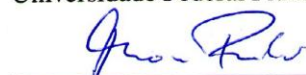
Prof. Dr. Mário Sarcinelli Filho
Universidade Federal do Espírito Santo
Orientador



Prof. Dr. Nelson Delfino d'Ávila Mascarenhas
Universidade Federal de São Carlos



Profa. Dra. Aura Conci
Universidade Federal Fluminense



Prof. Dr. Thomas Walter Rauber
Universidade Federal do Espírito Santo



Prof. Dr. Klaus Fabian Côco
Universidade Federal do Espírito Santo

*A minha mãe, meu falecido pai, minha paixão
e aos meus três tesouros.*

Agradecimentos

Gostaria de agradecer a todos os que me ajudaram!

À minha mãe que foi a mãe dos meus filhos enquanto eu estudava, ao meu irmão, Frank, e à minha cunhada, Marcia, que sempre que podiam cuidavam dos meus pimpolhos para que eu pudesse escrever artigos e varar noites.

Ao meu marido que varava as noites comigo (nem que seja só para fazer companhia), que me deu todo o suporte, que me ouvia discorrer horas sobre um assunto e que me apoiou nas minhas inúmeras variações de humor e continua casado comigo.

Agradeço aos meus orientadores, prof. Evandro e prof. Sarcinelli. Prof. Evandro sempre puxando na teoria, trazendo novos questionamentos. Paciência, filosofia e disposição sempre ocorrem ao falar dele. Prof. Sarcinelli nunca me deixou desanimar, sempre com palavras de conforto e incentivo. Acho que ele acredita mais em mim do que eu em mim mesma.

Meu muito obrigada para Klaus, que sempre me ajudou, com a maior boa vontade, seja explicando sua tese, seja explicando os mais variados assuntos.

Sou muito grata aos meus colegas do “laboratório sem janelas e sem placa”. Meu grupo de suporte: Patrick, Rossow, Anibal, Janayna e Jorge. Sempre que precisei, eu ligava para eles, chorava na orelha deles, conversava por horas e principalmente tomávamos café. Graças a eles, o tempo no laboratório foi divertido, mesmo brigando, sempre havia uma piada no final. Só eles, para entenderem o que eu estava passando.

Sumário

1	Introdução	16
1.1	Objetivo	21
1.2	A Abordagem Proposta	22
1.2.1	O Sistema Visual Humano	22
1.2.2	A Arquitetura Proposta	26
1.3	Metodologia de Avaliação dos Resultados	29
1.4	Contribuições	31
1.5	Limitações	32
1.6	Organização do Texto	33
2	Critério de Homogeneidade Integrando Descritor Multifractal e <i>J-image</i>	34
2.1	O método JSEG	34
2.2	A imagem do descritor multifractal	39
2.3	Trabalhos Relacionados	44
2.4	Arquiteturas Implementadas	46
2.4.1	Fractal- <i>only</i>	47
2.4.2	Fractal-JSEG	48
2.4.3	I-Frac	49
2.4.4	MM-Frac	51

2.5	Resultados Experimentais e Discussão	56
2.5.1	Comparação entre JSEG e Fractal- <i>only</i>	56
2.5.2	Comparação Fractal-JSEG x (JSEG e Fractal- <i>only</i>)	59
2.5.3	Comparação I-Frac x Fractal- <i>only</i>	63
2.5.4	Comparação I-Frac x (JSEG, Fractal-JSEG e Fractal- <i>only</i>)	65
2.5.5	Comparação MM-Frac x (JSEG, I-Frac e Fractal-JSEG)	66
2.5.6	Problemas em Aberto do MM-Frac	71
2.6	Considerações Finais	72
3	Integrando Crescimento de Regiões e Detecção de Bordas	74
3.1	Trabalhos Relacionados	75
3.2	Arquiteturas Implementadas	77
3.2.1	A Escolha do Detector de Bordas	78
3.2.2	KSS	81
3.2.3	KoSS	86
3.3	Resultados Experimentais e Discussão	89
3.3.1	BSDS	89
3.3.2	LHI	101
3.4	Considerações Finais	114
4	Conclusões Finais e Trabalhos Futuros	117
4.1	Trabalhos Futuros	119
4.2	Trabalhos Publicados	120
	Referências Bibliográficas	122

A	JSEG	133
A.1	Quantização de Cores	135
A.2	Segmentação Espacial	136
A.2.1	<i>J</i> -image	136
A.2.2	Crescimento de Regiões	139
B	<i>Precision</i>×<i>Recall</i> e ROC	141
C	BSDS	147
C.1	A Consistência das Anotações Humanas	148
C.2	O Cálculo das Métricas	150
D	Multifractal	153
D.1	O Método <i>Differential Box-Couting</i>	156

Lista de Tabelas

2.1	Características das Arquiteturas Implementadas.	46
2.2	Tamanho das janelas para cada escala do I-Frac.	51
2.3	Métricas calculadas pelo BSDS: referência humana, JSEG, Fractal- <i>only</i> , Fractal-JSEG, I-Frac e MM-Frac.	69
3.1	Tabela comparativa das métricas calculadas pelo BSDS nos métodos de detecção de bordas testados.	79
3.2	Tabela comparativa entre os métodos de detecção de bordas, mostrando a quantidade de imagens em que cada método (linha) obteve o maior valor da métrica (coluna).	79
3.3	Métricas <i>precision</i> , <i>recall</i> e <i>F-measure</i> calculadas pelo BSDS, comparação entre referência humana, JSEG, MM-Frac, detecção de bordas, KSS e KoSS.	96
A.1	Tamanho das janelas para cada escala.	139
B.1	Matriz de confusão para um classificador binário.	142

Lista de Figuras

1.1	Exemplos de imagens do BSDS (a,c,e,g,i) e suas respectivas segmentações feitas por anotadores humanos (b,d,f,h,j).	18
1.2	Um exemplo do BSDS (a) Imagem original 163085, (b-h) sete segmentações manuais feitas por diferentes anotadores humanos sobrepostos à imagem original.	20
1.3	Entrada e saída do sistema de segmentação automática e não-supervisionada.	22
1.4	Vista 3D de corte horizontal do cérebro mostrando o fluxo de informações visuais. Fonte: (Gunther, 2005).	23
1.5	Caminhos dorsal e ventral do processamento visual. Fonte: (Basso, 2005). .	23
1.6	Possíveis funções processadas pelos caminhos visuais. Fonte: extraído verbatim de (Kandel et al., 2000). Tradução dos ícones: color=cor, motion=movimento, depth=profundidade e form=forma.	24
1.7	Esquema simplificado dos caminhos P e M.	26
1.8	Arquitetura genérica da abordagem proposta.	28
1.9	Uma instância da arquitetura geral.	29
1.10	Processo de uso do BSDS.	30
1.11	Exemplo de imagem de entrada e seu referencial humano da base de dados DRIVE.	32
2.1	Resultados intermediários do método JSEG para a imagem 101085 do BSDS.	36
2.2	Resultados de segmentação do método JSEG.	37
2.3	Aplicação de diferentes filtros em um mosaico de texturas.	38

2.4	(a) Imagem 101085 da base de dados do BSDS; (b) Imagem do descritor fractal para janela de tamanho 9×9 ; (c) gráfico 2D da linha 200 da imagem (a); (d) gráfico em 2D da linha 200 da imagem (b); (e) Imagem do descritor fractal para janela de tamanho 17×17 ; (f) Imagem do descritor fractal para janela de tamanho 33×33 ; (g) gráfico 2D da linha 200 da imagem (e); (h) gráfico em 2D da linha 200 da imagem (f).	41
2.5	À esquerda apresentam-se as imagens dos descritores fractais normalizados, (o tamanho da janela associada se encontra abaixo de cada imagem). À direita, apresentam-se os gráficos 2D da linha 200 da imagem à esquerda. .	42
2.6	(a) Referência Humana de 101085. (b) Gráfico 2D com as curvas da linha 200 da imagem em escala de cinza 101085 em azul (c) Gráfico 2D com as curvas da linha 200 da imagem do descritor multifractal com janela 9×9 em azul. A curva em vermelho de (b) e (c) corresponde à linha 200 da imagem da referência humana.	43
2.7	Arquitetura do Fractal-only.	47
2.8	Arquitetura do Fractal-JSEG.	49
2.9	Arquitetura do I-Frac.	50
2.10	Arquitetura de MM-Frac.	52
2.11	Espectro de potência de uma imagem em 3D.	53
2.12	Espectro de potência de uma imagem em 2D.	53
2.13	Exemplos de imagens com valores de α próximos de 1.	54
2.14	Exemplos de imagens com valores de α próximos de 3.	55
2.15	Exemplos característicos com texturas complexas nos quais o Fractal-only apresentou valores de <i>F-measure</i> maiores que o JSEG.	57
2.16	Exemplos característicos com similaridade de cor-textura de diferentes componentes, nos quais o Fractal-only apresentou valores de <i>F-measure</i> menores que o JSEG.	58
2.17	Exemplos característicos onde as anotações humanas foram detalhadas e nos quais o JSEG apresentou valores de <i>F-measure</i> maiores que o Fractal-only. .	59
2.18	Imagens associadas a imagem 167062.	59

2.19	Exemplos nos quais o Fractal-JSEG melhorou a métrica <i>F-measure</i> com relação aos métodos JSEG e Fractal-only. (a) Imagem de entrada (b) Anotação humana (c) resultado do JSEG (d) resultado do Fractal-only (e) resultado do Fractal-JSEG	61
2.20	Histogramas das métricas dos métodos JSEG \times Fractal-only \times Fractal-JSEG.	62
2.21	Exemplos com similaridade de cor-textura dos componentes nos quais o I-Frac obteve maiores valores de <i>F-measure</i> que Fractal-only.	63
2.22	Resultados para a imagem 196073 (a) Resultado do Fractal-JSEG sobreposto à imagem original (b) Resultado do I-Frac sobreposto à imagem original	64
2.23	Exemplos de imagens com texturas complexas.	64
2.24	Exemplos em que o método I-Frac aumentou os valores de <i>F-measure</i> . (a) Imagem de entrada (b) Anotação humana (c) JSEG (d) Fractal-only (e) Fractal-JSEG (f) I-Frac.	66
2.25	Histogramas das métricas dos métodos JSEG \times Fractal-only \times Fractal-JSEG \times I-Frac.	67
2.26	Exemplos de resultados do método MM-Frac, na ordem: (a) Imagem de entrada (b) Referência humana e as imagens resultados dos métodos (c) JSEG (d) Fractal-JSEG (e) I-Frac (f) MM-Frac.	68
2.27	Histogramas das métricas dos métodos JSEG \times Fractal-JSEG \times I-Frac \times MM-Frac.	70
2.28	Exemplo característico do problema em aberto do MM-Frac. (a) Imagem de entrada (b) Referência humana e as imagens resultados dos métodos (c) JSEG (d) Fractal-only (e) Fractal-JSEG (f) I-Frac (g) MM-Frac.	71
2.29	Exemplo de generalização de textura complexa do I-Frac.	72
3.1	Arquitetura geral da estratégia de integração.	77
3.2	Gráficos de <i>Precision</i> \times <i>Recall</i> dos detectores de borda.	80
3.3	Gráficos de <i>Precision</i> \times <i>Recall</i> do Gradiente Morfológico aplicado em imagem em escala de cinza e aplicado em imagem colorida com pré-processamento via filtro de Kuwahara.	81
3.4	Arquitetura do KSS.	82

3.5	Exemplos de imagens para um baixo e alto valor de $\limiar_{bordafraca}$. (a) e (b) imagens de entrada; (c) e (d) imagens provenientes do detector de bordas; (e) e (f) histogramas das imagens (c) e (d), respectivamente, e (g) e (h) imagens referentes ao mapa-de-bordas-fracas.	84
3.6	Imagens de entrada e saída do KSS. Imagens (a) e (b) resultantes do detector de bordas; (c) e (d) resultantes do MM-Frac; (e) e (f) resultantes do KSS e, (g) e (h) anotações humanas.	85
3.7	KSS \times KoSS: artefatos no resultado do KSS, mas não no do KoSS.	88
3.8	Exemplos comparativos (a) Imagem de entrada (b) Referência humana e as imagens resultados dos métodos (c) detecção de borda (d) MM-Frac (e) KSS (f) KoSS.	91
3.9	Exemplos comparativos (a) Imagem de entrada (b) Referência humana e as imagens resultados dos métodos (c) detecção de borda (d) MM-Frac (e) KSS (f) KoSS.	92
3.10	Exemplos comparativos (a) Imagem de entrada (b) Referência humana e as imagens resultados dos métodos (c) detecção de borda (d) MM-Frac (e) KSS (f) KoSS.	93
3.11	(a) Imagem de entrada (b) Referência humana e as imagens resultados dos métodos (c) detecção de borda (d) MM-Frac (e) KSS (f) KoSS.	94
3.12	Resultados do KSS/KoSS que apresentaram $F\text{-measure}$ menor que que ambos os métodos MM-Frac e detecção de borda (a) Imagem de entrada (b) Referência humana e as imagens resultados dos métodos (c) detecção de borda (d) MM-Frac (e) KSS (f) KoSS.	95
3.13	Gráfico $\text{precision} \times \text{recall}$ dos métodos gradiente morfológico, KoSS, JSEG e MM-Frac.	97
3.14	Histogramas das métricas dos métodos KoSS \times MM-Frac \times detector de bordas (na legenda dos gráficos como “Edge”).	98
3.15	Imagens 296059 e 302008 com seus referenciais humanos.	100
3.16	Imagem 12084 com seus referencial humano.	100
3.17	Imagens 210088 com seu referencial humano.	101
3.18	Imagem 69040 com seu referencial humano.	101

3.19	Exemplos do PASCAL VOC, onde a coluna esquerda (a) mostra as imagens de testes do banco de dados e a coluna direita (b) mostra o referencial humano das imagens à sua esquerda.	103
3.20	Resultado desejado em pesquisas utilizando o MSRC.	103
3.21	Resultados para a imagem do LHI (tamanho 1400×838).	106
3.22	Resultados para a imagem do LHI (tamanho 1400×838).	107
3.23	Resultados para a imagem do LHI (tamanho 1400×838).	108
3.24	Resultados para a imagem do LHI (tamanho 1400×838).	109
3.25	Exemplos do LHI, imagens resultante sobrepostas à imagem original. . . .	110
3.26	(a) Imagem original (b) referencial humano e os resultados dos métodos (c) JSEG (d) MM-Frac (e) detecção de bordas (f) KoSS (g) KoSS com pós-processamento.	111
3.27	(a) Imagem original (b) referencial humano e os resultados dos métodos (c) JSEG (d) MM-Frac (e) detecção de bordas (f) KoSS (g) KoSS com pós-processamento.	113
3.28	Exemplos do LHI, imagem resultante sobreposta às imagem original. . . .	114
4.1	Arquitetura genérica da abordagem proposta.	117
A.1	Arquitetura em dois estágios do JSEG. (a) visão básica (b) visão mais detalhada de (a) (c) apresentando algoritmo de segmentação espacial com o controle de quantidade de escalas.	134
A.2	Exemplos de três mapas-de-classes. Fonte: Deng e Manjunath (2001) . . .	136
A.3	“Melhor” segmentação dos mapas-de-classe 1 e 3. Fonte: Deng e Manjunath (2001)	138
B.1	A matriz de confusão e as métricas que compõe as curvas ROC e <i>precision</i> \times <i>recall</i> . Fonte: Pres (2008).	143
B.2	Dois algoritmos sendo comparados nos espaços ROC e <i>Precision</i> \times <i>Recall</i> . Fonte: Davis e Goadrich (2006)	144

B.3	Curvas ROC e <i>Precision</i> × <i>Recall</i> em diferentes distribuições de classes (a) curvas ROC 1:1 (b) curvas <i>Precision</i> × <i>Recall</i> 1:1 (c) curvas ROC 1:10 (d) curvas <i>Precision</i> × <i>Recall</i> 1:10 . Fonte: Fawcett (2006).	145
B.4	Conjuntos dos pixels retornados pelo método, que podem ser relevantes ou irrelevantes.	146
C.1	Exemplo das diferentes formas de apresentação para o anotador humano. Fonte: Martin (2002)	148
C.2	Exemplos de segmentações a sua árvore de percepção associada. Fonte: Martin (2002)	149
C.3	Exemplo de inconsistência entre anotações humanas. Fonte: Martin (2002)	150
C.4	Exemplo de um grafo bipartido simples.	150
C.5	Esquema simplificado dos passos para a comparação entre duas segmentações. Fonte: Martin et al. (2004)	151
D.1	Fractal de Mandelbrot.	154
D.2	Fractais estatísticos.	154
D.3	Exemplo do algoritmo DBC.	157
D.4	Singularidades do tipo degrau e cúspide.	158
D.5	Imagem do descritor multifractal.	159

Resumo

A segmentação de cenas naturais é uma tarefa importante em processamento de imagens e visão computacional, com aplicações em diversas áreas, tais como navegação robótica e reconhecimento de objetos. Entretanto, a etapa de segmentação pode se tornar extremamente complicada devido à imensa variabilidade de cor, iluminação e texturas que se manifestam em uma imagem. Ou seja, é muito difícil desenvolver uma abordagem que consiga segmentar satisfatoriamente todas as nuances de uma cena projetada numa imagem.

Esta Tese propõe um novo método não supervisionado e automático para detecção de contornos em imagens naturais coloridas, consistindo de dois níveis de integração, ou processos sequenciais de dois estágios. No primeiro estágio, duas diferentes técnicas para medir a homogeneidade cor-textura em um método de crescimento de regiões são combinadas por dois diferentes algoritmos de controle. Um algoritmo de controle é baseado em uma função local e outro é baseado em uma propriedade estatística global (a forma do espectro de potência da imagem que está sendo analisada). Uma medida de homogeneidade é o *J-value* (fornecido pelo algoritmo JSEG) e a segunda medida é o descritor multifractal. O primeiro estágio fornece a extração de regiões. Na segunda integração, a informação de bordas é extraída por um método clássico, e integrada com a informação de região. Este processo elimina os contornos falsos do mapa de regiões, guiado pelo mapa de bordas, e reduz o ruído do mapa de bordas, agora guiado pelo mapa de regiões, tirando, assim, proveito da natureza complementar de cada método. Além disso, integram-se os dois mapas em um único resultado final, reforçando as informações coincidentes de ambas as entradas.

Cada fase de integração melhora, progressivamente, a detecção de contornos. Experimentos com imagens retiradas de um banco de dados extenso e consolidado de cenas naturais coloridas (“The Berkeley Segmentation Dataset and Benchmark”) sugerem que os resultados da abordagem deste trabalho são mais próximos da percepção humana que os métodos individuais, tanto quantitativamente quanto qualitativamente falando.

Abstract

Segmentation of natural scenes is an important task in image processing and computer vision, with applications in several areas such as robot navigation and object recognition. However, the segmentation can become extremely complex due to the huge variability of color, lighting and textures found in an image. In other words, it is very difficult to develop an approach that can successfully segment all changes in a scene.

This Thesis proposes a new unsupervised and fully automatic method for boundary detection in natural color images, consisting of two levels of integration, or two-stage sequential processes. In the first stage, two different techniques to measure color-texture homogeneity in a region-growing method are combined by two different control algorithms. One control algorithm is based on a local function and the other is based on a global statistical property (the shape of the power spectrum of the image being analyzed). One homogeneity measure is the J-value (provided by the JSEG algorithm) and the second measure is a multifractal descriptor. This first stage performs region extraction. In the second integration, edge information is extracted by a classical method, and integrated with region information. This process eliminates false boundaries in the region map, guided by the edge map, and reduces the noise in the edge map as well, now guided by the region map, thus taking advantage of their complementary nature. Furthermore, it integrates the two maps into a single final result, enhancing the coincident information of both maps.

Each phase of integration improves, progressively, the detection of the boundaries. Experiments on a large and consolidated dataset of natural color images (“The Berkeley Segmentation Dataset and Benchmark”) suggest that the results for the approach here proposed are closer to the human perception than the individual methods, quantitatively and qualitatively speaking.

Capítulo 1

Introdução

O reconhecimento automático de objetos em imagens digitais por sistemas computacionais possui inúmeras aplicações, tais como indexação e recuperação automáticas de fotos e vídeos, uso em sistemas de navegação de robôs, monitoramento de poluição, sistemas de segurança e sistemas de inspeção de qualidade de produtos (Jain, 1989).

Uma etapa prévia ao reconhecimento de objetos é a segmentação de regiões, que consiste em prover uma decomposição da imagem em diversas sub-regiões, que guardam, cada uma delas, alguma descrição comum. O objetivo da segmentação é simplificar e/ou alterar a representação da imagem para algo que seja mais significativo e fácil de se analisar. A “detecção de fronteiras” é a identificação das curvas divisórias entre as diversas sub-regiões da imagem (Gonzalez e Woods, 2001).

Um sistema de segmentação pode ser classificado como manual, semiautomático ou automático, dependendo do grau de intervenção humana (Ventura, 2009). A segmentação é dita manual quando um indivíduo humano define todas as fronteiras das regiões da imagem. Nos algoritmos de segmentação semiautomáticos, a intervenção manual é utilizada para fornecer pontos característicos da estrutura a ser segmentada, ou para delimitar uma região onde ela é encontrada, ou ainda para indicar a quantidade de regiões a serem segmentadas. Já na segmentação automática não há qualquer intervenção humana.

A segmentação pode ainda ser classificada como supervisionada ou não-supervisionada (Jain et al., 2000). A diferença é que os sistemas de segmentação supervisionados passam por uma fase de treinamento, recebendo imagens previamente analisadas e rotuladas, definindo-se um padrão e um conhecimento a priori daquilo que vai ser segmentado na fase de testes. Portanto, na fase de treinamento o sistema deve ajustar seus parâmetros e se adaptar a imagens com características similares e na fase de testes, a entrada é identificada como um membro de uma das classes pré-definidas do treinamento. Na abordagem não-supervisionada não

há classes definidas a priori, as classes são aprendidas com base na similaridade dos padrões durante a fase de testes. De acordo com a entrada, os limites de decisão são construídos e o padrão de entrada é associado a uma classe até então desconhecida. Resumindo, o objetivo da classificação não-supervisionada é encontrar agrupamentos naturais, ou “clusters”, em dados multidimensionais com base nas semelhanças entre os padrões.

A segmentação automática e não-supervisionada de cenas naturais é um problema complexo, pois não se conhece, a priori, quais tipos de regiões existem em uma imagem, nem quantas são as regiões. Ou seja, não há nenhuma informação além da própria imagem digital. A imagem pode conter regiões uniformes (apresentando uma única cor), regiões sombreadas com gradação suave de cor e regiões com texturas (Tu e Zhu, 2002). As texturas podem ser as mais variadas, podendo ser sintéticas, apresentando uma repetição de padrões estruturais, ou aleatórias com informação estatística não-aleatória, e cada textura pode apresentar diferentes granularidades.

A estratégia de um sistema de segmentação automática e não-supervisionada de cenas naturais coloridas se resume a uma forma de distinguir quando duas regiões adjacentes são diferentes em uma imagem digital, que é a sua única informação. Um ser vivo recebe outras informações obtidas através de outros sensores para efetuar a segmentação. Por exemplo, o ser humano possui dois olhos, duas imagens, que adicionam informações sobre a distância dos elementos da cena e o ser humano.

Através de cinco exemplos, mostram-se, a seguir, algumas dificuldades inerentes a segmentação de imagens. Na Figura 1.1, a coluna esquerda apresenta alguns exemplos de imagens retiradas do “The Berkeley Segmentation Dataset and Benchmark” (BSDS) (Martin et al., 2001), e a coluna direita mostra as suas respectivas imagens de segmentações feitas por anotadores humanos. As imagens da coluna direita apresentam segmentações sobrepostas de no mínimo cinco anotadores humanos, assim, quanto mais evidente a curva, maior foi a coincidência de marcação entre os anotadores. Cada linha desta Figura 1.1 apresenta um aspecto diferente a ser analisado:

1. a primeira linha apresenta uma mudança suave de iluminação em seus quatro cantos, característica que não seria segmentada por nenhum anotador humano;
2. a segunda linha mostra uma onça em um galho. Deve-se notar a complexidade da textura de seu pelo: embora exista repetição de padrão, esta repetição se apresenta em tamanhos e orientações diferentes. Apesar desta complexidade, a percepção humana a entende como um único elemento na foto, e não segmenta cada detalhe de sua textura (pode-se conjecturar que quando há muita informação numa imagem, o ser humano tende a usar apenas as informações mais “grosseiras” de sua percepção, não se atendo aos detalhes);

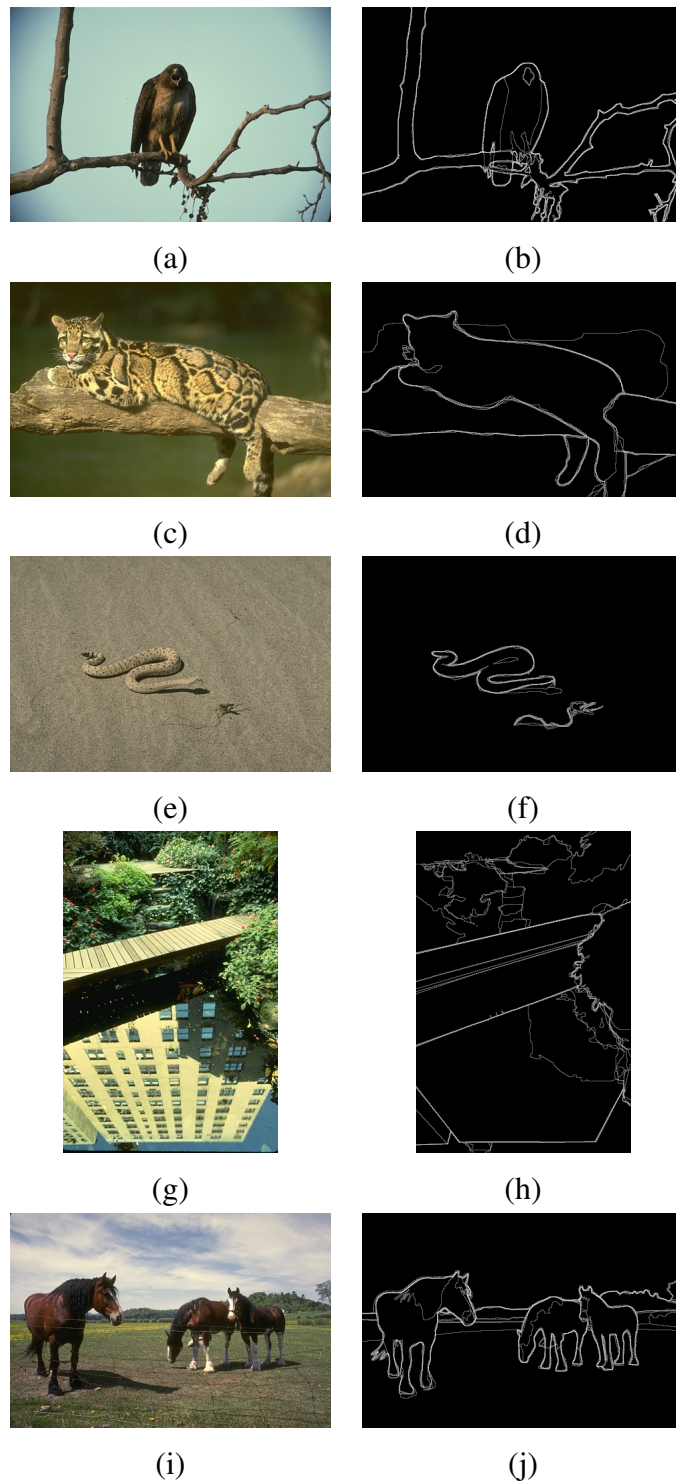


Figura 1.1: Exemplos de imagens do BSDS (a,c,e,g,i) e suas respectivas segmentações feitas por anotadores humanos (b,d,f,h,j).

3. a terceira linha mostra uma cobra no deserto. Observa-se que o elemento central e o fundo apresentam quase a mesma cor, uma camuflagem da natureza. Essa imprecisão da definição das bordas é um obstáculo para a segmentação;
4. na quarta linha, tem-se uma mistura de texturas artificiais e naturais, apresentando características diferentes. O prédio e a ponte apresentam texturas artificiais, isto é, possuem padrões estruturais determinísticos, repetitivos e periódicos, ao contrário da textura natural da folhagem, que é aleatória;
5. o último exemplo mostra três cavalos no campo. Note-se que os anotadores humanos não segmentam a região das sombras dos cavalos, principalmente a sombra do cavalo mais à esquerda, que ocupa um bom espaço do chão. Sistemas automáticos tenderão a segmentar esta sombra, dado que a região de sombra e a região de grama possuem características diferentes de cor.

Seres humanos usam informações cognitivas superiores, como o reconhecimento de objetos e texturas, e também conhecimento prévio quanto à forma de objetos e seres. Na segmentação manual da imagem da cobra (Figura 1.1f), quatro seres humanos reconheceram o que era a cabeça da cobra e não segmentaram a sombra, e apenas um ser humano segmentou o contorno da cobra e a sombra.

Mesmo se tendo um sistema que processe a segmentação, surge outra questão: como avaliar a qualidade da segmentação proposta, dada uma imagem segmentada? Até o momento, o instrumento mais eficiente para detectar e discriminar regiões é o olho humano, servindo, na maioria das vezes, como referência para comparação dos algoritmos utilizados. No entanto, uma característica da percepção visual humana é que ela é ambígua e imprecisa. Imagens naturais possuem tantos elementos que muitas vezes, mesmo entre seres humanos, não há consenso na definição de quais são as regiões de uma imagem.

Um exemplo de segmentação não consensual é mostrado na Figura 1.2, o qual também é retirado da base de dados BSDS. A Figura 1.2a é uma cena natural contendo três passarinhos num ninho, e as Figuras 1.2b a 1.2h são as segmentações manuais feitas por diferentes anotadores humanos, cada uma sobreposta na imagem original.

Pode-se constatar pelos resultados das segmentações que mesmo entre os anotadores humanos não há consenso, embora exista consistência entre as versões. As versões do quarto e sétimo anotadores (Figuras 1.2e e 1.2h) identificam os bicos e os olhos dos passarinhos, porém não são contornos idênticos. O quarto anotador é bem detalhista ao dividir as penas de cada passarinho em duas partes, uma relativa à textura das penas do corpo, que aparenta ser mais espessa, e a outra relativa à penugem ao longo de suas costas, com aparência

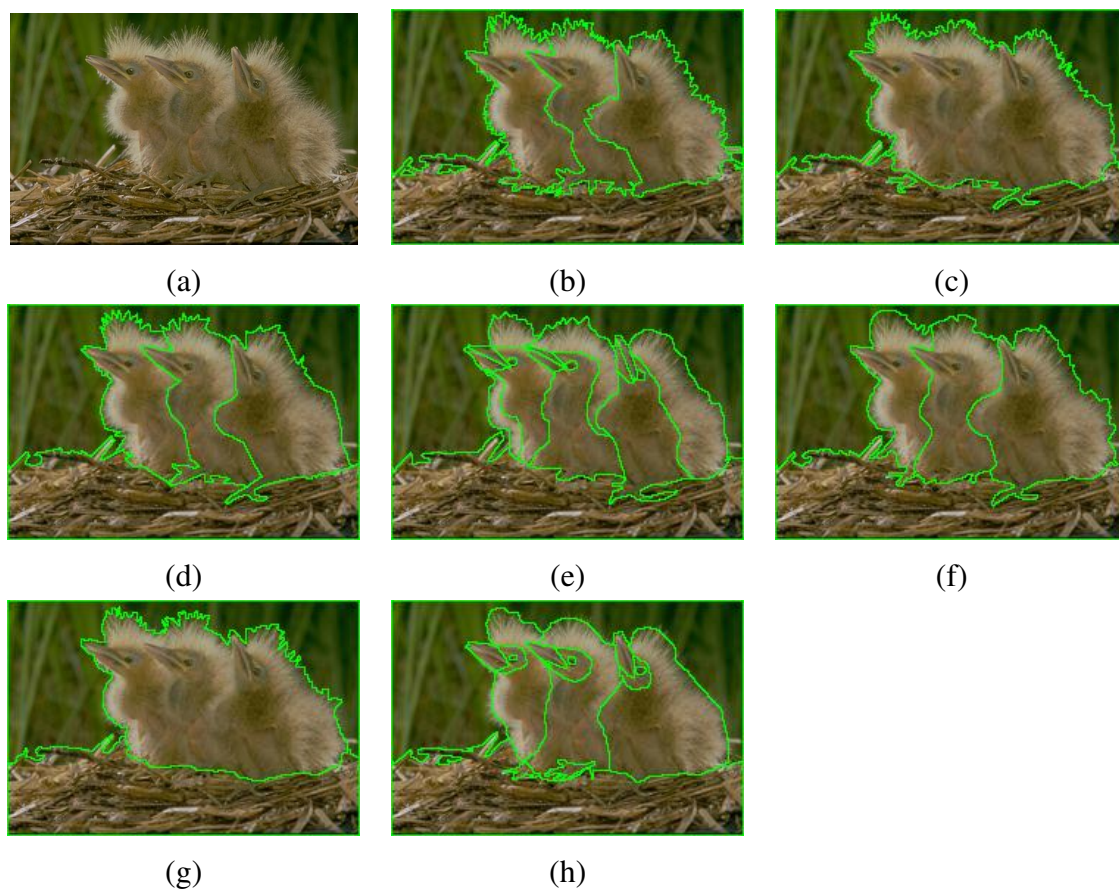


Figura 1.2: Um exemplo do BSDS (a) Imagem original 163085, (b-h) sete segmentações manuais feitas por diferentes anotadores humanos sobrepostas à imagem original.

mais fina. Há dois anotadores (Figuras 1.2c e 1.2g) menos detalhistas, que não apresentam linhas de distinção entre os corpos de cada passarinho. Os outros cinco anotadores (Figuras 1.2b, 1.2d, 1.2e, 1.2f e 1.2h) distinguem cada passarinho, porém cada contorno é desenhado diferentemente.

Nas segmentações, os resultados comuns foram as identificações da divisão do ninho e do contorno geral dos passarinhos. No entanto, é possível verificar que cada anotador desenha uma linha de contorno, cobrindo os três passarinhos, de uma forma diferente. Há anotadores, como os das Figuras 1.2b e 1.2c, que desenharam esta divisão de forma mais irregular, bem diferente do contorno suave do anotador da Figura 1.2h. Da mesma forma, a linha divisória entre o ninho e os passarinhos apresenta diferenças em cada anotação.

Estas diferenças entre os resultados das segmentações humanas podem ser devido à complexidade das texturas (penugem dos passarinhos, vegetação ao fundo e ninho), à forma complexa dos componentes da cena (não há uma clara fronteira entre estas diferentes componentes), ou ao desconhecimento prévio da forma dos objetos. Assim, a segmentação manual de uma imagem não é única: diferentes indivíduos irão gerar diferentes resultados, um mesmo

indivíduo pode gerar resultados diferentes em tempos diferentes, e um mesmo indivíduo pode gerar resultados diferentes dependendo da aplicação.

Um ser humano faz uma interpretação semântica da cena de acordo com a aplicação. Assim, um especialista em observação de passarinhos pode estar interessado apenas nos animais e não estar interessado no restante da imagem, enquanto outro especialista pode estar interessado apenas no ninho e não nos outros componentes da imagem. Esta ambiguidade da semântica torna impossível (ainda) que o computador automaticamente agrupe as várias regiões homogêneas em um único objeto semântico significativo. Por outro lado, mesmo que o ser humano seja responsável pela definição semântica, um sistema de segmentação automático fornecerá uma primeira aproximação que auxiliará o ser humano a encontrar as fronteiras entre as regiões, que, em geral, é um trabalho manual maçante.

Devido à variedade de dificuldades, o estudo de segmentação de cenas naturais continua sendo tópico de grande interesse. Embora já se estenda por mais de três décadas, e apesar da quantidade de trabalhos propostos, ainda não há uma técnica que resolva todos os desafios de forma eficiente (Gonzalez e Woods, 2001; Fauzi e Lewis, 2003; Cheng et al., 2001; Conci e Nunes, 2007).

1.1 Objetivo

O objetivo principal deste trabalho é propor uma solução para o problema de segmentação automática e não-supervisionada de cenas naturais coloridas. Esta proposta compreende estágios sequenciais em que cada estágio aborda a integração de informações complementares.

A única entrada do sistema será a imagem digital colorida, e a única saída será uma imagem indicando as fronteiras das regiões, conforme a Figura 1.3, onde a imagem de entrada é interpretada como um conjunto de três matrizes, que é a representação comum de imagens digitais coloridas.

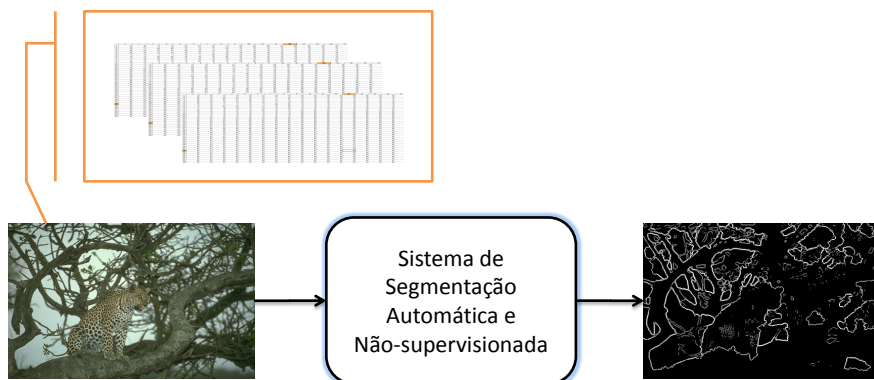


Figura 1.3: Entrada e saída do sistema de segmentação automática e não-supervisionada.

1.2 A Abordagem Proposta

Uma possível forma de encurtar o caminho do desenvolvimento de sistemas de segmentação de imagens naturais é aprender com o sistema biológico humano de visão (Komati et al., 2003), sobre o comportamento dos seus neurônios e sobre como se dá a percepção visual a partir do comportamento destes neurônios. Visão é o mais poderoso e complicado dos sentidos do ser humano, e o conhecimento acerca da visão biológica humana não é completo e detalhado (Olshausen e Field, 2005). Isto tem motivado muitos pesquisadores a propor teorias sobre como se dá o processo de visão biológica, para posterior comparação com o comportamento visual biológico observável. Tais teorias têm evoluído ao longo dos anos, baseadas na crescente compreensão do sistema visual humano (SVH), através do qual se “percebem” as posições e propriedades dos objetos, assim como as relações entre si e as leis físicas do ambiente que os cerca.

1.2.1 O Sistema Visual Humano

Inicia-se esta seção com uma descrição sucinta do conhecimento estabelecido acerca do fluxo da informação visual (Kandel et al., 2000). O sistema tem início com a captação das imagens através dos olhos, por meio da retina, localizada no fundo do olho, transformando a informação visual em sinais elétricos. Na Figura 1.4 observam-se as fibras do nervo óptico chegando ao tálamo (*thalamus*) num primeiro estágio, e se encaminhando para o córtex visual, num segundo estágio. Além disso, há cruzamento de informações da retina esquerda e direita no tálamo.

Dentro do tálamo encontra-se o núcleo geniculado lateral (NGL), onde os sinais canalizados através do nervo óptico são divididos em duas vias visuais distintas: o caminho

magnocelular (M) e o caminho parvocelular (P). Do NGL, os axônios seguem em direção ao córtex visual, onde ocorre o processamento das informações visuais, ou seja, as características de forma, profundidade, movimento e cor são analisadas. Ele é dividido em áreas: V1, V2, V3, V4 e V5 (também conhecido como MT), conforme se pode ver na Figura 1.5.

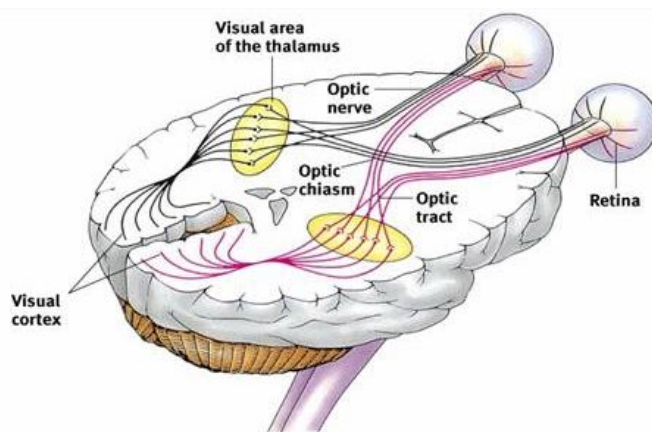


Figura 1.4: Vista 3D de corte horizontal do cérebro mostrando o fluxo de informações visuais. Fonte: (Gunther, 2005).

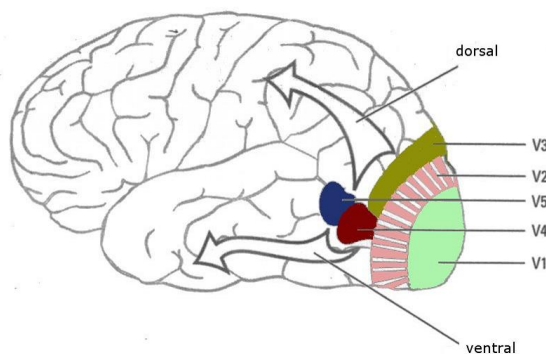


Figura 1.5: Caminhos dorsal e ventral do processamento visual. Fonte: (Basso, 2005).

No caminho magnocelular as informações saem das células do tipo M do NGL, entram em V1, depois vão para V2, de V2 vão para V3 e V5, e, finalmente, seguem para o lobo parietal (caminho dorsal). Células em MT respondem a movimentos de rotação de cenas, tais como os que são causados pelos movimentos de cabeça. No geral, este caminho apresenta alta sensibilidade para contraste, baixa resolução espacial, baixa sensibilidade a cores e alta resolução temporal, tornando-o capaz de detectar movimentos e permitindo habilidades de respostas rápidas e adaptativas, embora transientes, a determinados estímulos, mas é pobre no que se refere à compreensão de contornos ou discernimento de bordas. Schiller e colegas (Schiller et al., 1980) descobriram que este caminho também apresenta capacidade para percepção de profundidade, mas não para objetos estacionários. Uma lesão neste caminho leva à deficiência de percepção de movimentos e perseguição suave de alvos.

No caminho parvocelular as informações saem das células P do NGL, vão para os neurônios de V1, seguem para V2, depois para V4 e, finalmente, para o córtex temporal inferior (caminho ventral). Este caminho é composto por neurônios com características de baixa sensibilidade ao contraste, alta resolução espacial, baixa resolução temporal e alta sensibilidade a cantos e bordas, o que o torna capaz de analisar detalhes estacionários do mundo real e de detectar formas. Ele é sensível à cor e a detalhes finos. Os neurônios que compõem este caminho também contribuem para a percepção de profundidade. Lesões neste caminho causam dificuldades de reconhecimento de objetos e pessoas.

Ainda não existe teoria estabelecida sobre como as informações sobre movimento, cor, profundidade e forma são organizadas de forma perceptualmente coesa. Uma possível explicação é que as imagens visuais são processadas em vias paralelas, onde cada via é formada por grupos independentes de células com funções diferentes, e depois há uma combinação destas características (movimento, profundidade, forma e cor). O mecanismo pelo qual o cérebro coordena a associação da informação processada de forma independente ainda é indefinido, e é chamado de “*binding mechanism*” (Kandel et al., 2000). Qualquer caso do cérebro associar duas ou mais características extraídas separadamente do mesmo objeto pode ser chamada de “*binding*” (Holcombe, 2009).

Um possível mapeamento de características nos caminhos visuais é mostrado na Figura 1.6. Esta imagem foi extraída de forma verbatim, e, logo, os termos estão em inglês. Supõe-se que o caminho magnocelular processa características de movimento, profundidade e informação espacial, e o caminho parvocelular processa informações de forma e cor.

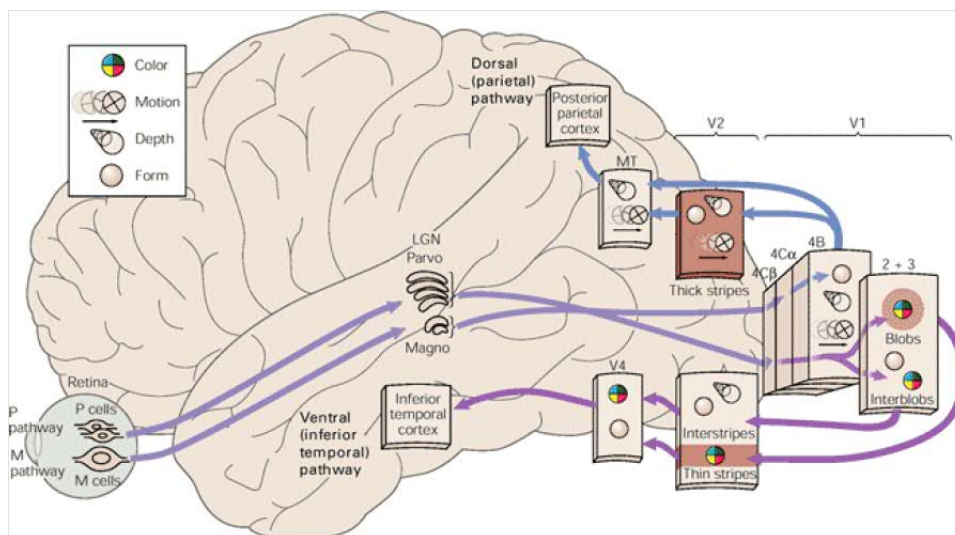


Figura 1.6: Possíveis funções processadas pelos caminhos visuais. Fonte: extraído verbatim de (Kandel et al., 2000).

Tradução dos ícones: color=cor, motion=movimento, depth=profundidade e form=forma.

Outra teoria é a de Treisman e colegas (Treisman et al., 1977), que sugere que há dois processos distintos e sequenciais envolvidos na percepção visual. Primeiro, o processo “*pre-attentive*” (pré-atencional), que está preocupado apenas com a detecção de objetos. Este processo rapidamente examina a textura global de uma imagem e concentra-se na distinção entre figura e fundo através da codificação em paralelo das propriedades elementares: cor, orientação, tamanho, direção ou movimento. Neste ponto, a variação em uma propriedade simples pode ser percebida como um contorno, mas combinações complexas de propriedades elementares não são detectadas. Este agrupamento inicial de itens é seguido por um processo de “atenção” que seleciona e destaca as características consideradas mais importantes de um objeto, integrando-os em uma “mapa de saliência” (*saliency map*), e com isso desviando e direcionando a atenção para as áreas mais evidentes.

Uma outra teoria bem extensa é a do pesquisador Grossberg (Grossberg e Mingolla, 1985; Grossberg, 2000a,b, 2006), fruto de uma ampla pesquisa, iniciada na década de 1960. O modelo do SVH proposto por Grossberg é bem completo, e envolve explicações acerca da percepção humana, sugerindo soluções para a percepção 3D, visão binocular, dominância ocular, oclusão de bordas, aprendizado e memória, dentre outros desafios. A seguir se resumem algumas das hipóteses do modelo de Grossberg, com foco nas características associadas à detecção de contornos:

1. segmentações de imagens geradas pelo modelo não são resultados de treinamento em exemplares de imagens;
2. teoria “*Hierarchical Resolution of Uncertainty*”, onde incertezas devem ser superadas usando mais de uma fase de transformação para cada caminho, como, por exemplo, no processamento dos caminhos P ou M. As incertezas podem ser interpretadas tendo em conta o fato de que o resultado de cada estágio, ou unidade funcional do sistema, pode gerar resultados com falhas ou com graus diferentes de confiabilidade, que podem ser corrigidos gradativamente, estágio a estágio;
3. o cérebro possui uma organização de processamentos paralelos e de naturezas complementares, teoria de nome “*Complementary Computing*”. A idéia é que existem “pares de processamento de fluxos complementares”, onde cada caminho exhibe vantagens e desvantagens. Por causa dessa complementariedade, acredita-se que as interações entre estes caminhos sobrepujem suas deficiências complementares;
4. Grossberg insere em seu modelo o conceito “*The Two Streams Hypothesis*”, originalmente proposto por Mishkin e Ungerleider (1982). A hipótese considera que cada um dos caminhos das informações visuais se projeta para áreas diferentes, e, logo, processam atributos distintos. Simplificadamente, o caminho M é especializado em

responder à pergunta “Onde” (*stream* “Where”), sendo usado para localização de objetos, enquanto o caminho P é especializado em responder à pergunta “O que” (*stream* “What”), sendo usado para identificação de objetos;

5. a teoria neural, conhecida como FACADE (*Form-And-Color-And-Depth*), propõe que no caminho P uma das ramificações dentro do córtex visual faz o processamento de borda e a outra o de superfície (Figura 1.7). Seguindo a teoria “*Complementary Computing*”, as interações entre o processamento de borda e de superfície selecionam o melhor agrupamento de bordas entre as várias possibilidades de interpretação da cena. Através de suas propriedades complementares, o processamento de um caminho guia os sinais do outro caminho. A Figura 1.7 é apenas uma interpretação simplificada e esquemática dos caminhos paralelos. A simplificação é tal que não mostra a troca de informações entre os dois caminhos (embora se deva registrar que não há conhecimento estabelecido de como é feita esta comunicação entre os caminhos).

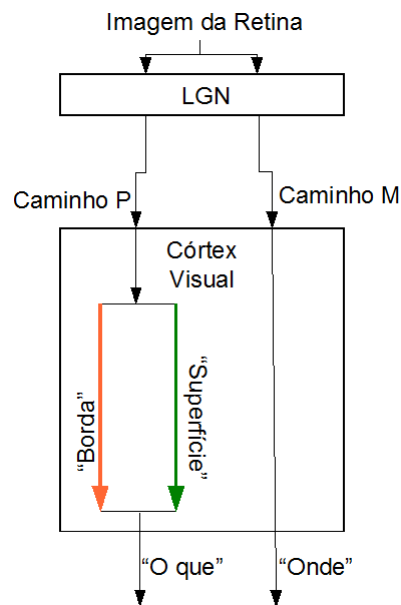


Figura 1.7: Esquema simplificado dos caminhos P e M.

1.2.2 A Arquitetura Proposta

Dar-se-á, nesta seção, uma visão geral da abordagem proposta, as escolhas feitas e as suposições assumidas. Inicia-se listando as seguintes pressuposições:

1. cada imagem de entrada contém um conjunto finito de regiões homogêneas, definidas com base nas suas cores e texturas;

2. segmentações de imagens de cenas naturais coloridas não são resultados do treinamento com exemplares de imagens. Abordagens não-supervisionadas são mais robustas ao dilema “polarização” \times variância (Duda et al., 2001), uma vez que não dependem de uma sintonia de parâmetros. Sistemas que necessitam de fase de treinamento podem apresentar *underfitting*, ou seja, podem ficar polarizados para um determinado tipo de resposta, apresentando pouca ou nenhuma variância. Ou podem apresentar *overfitting*, ou seja, tender a ter maior variabilidade nas respostas, mas não generalizar a resposta. Dessa maneira, essas duas características são tipicamente conflitantes, ou seja, a diminuição da polarização implica em aumento da variância e a diminuição da variância implica em aumento da polarização. O problema do treinamento é encontrar um equilíbrio entre as duas. Se o sistema for supervisionado, então necessitará de treinamento para ajuste de parâmetros, de maneira que qualquer mudança do banco de imagens de teste implicará em retreinamento do sistema. Devido à imensa variabilidade de cor, iluminação e texturas que se manifestam em imagens ao ar livre, sugere-se que um sistema de segmentação deve ser flexível e não ter uma fase de treinamento. Neste caso, o sistema fica também independente das imagens de teste;
3. são necessários mais de um estágio para resolução do problema, tal como a hipótese do “*Hierarchical Resolution of Uncertainty*”; e
4. a extração de características é feita de forma paralela e separada. Mais ainda, trabalhar com a combinação de duas informações complementares provenientes da mesma imagem de entrada é uma boa abordagem para melhoria de resultados, tal qual expressa no “*Complementary Computing*”.

Com isso, a arquitetura geral da abordagem proposta pode ser resumida como na Figura 1.8, onde há mais de um estágio (pressuposição 3), e em cada estágio há a integração de informações complementares (pressuposição 4).

Para testar a arquitetura proposta, desenvolveu-se uma instância da mesma, que foi testada tanto quantitativamente quanto qualitativamente. Neste sistema, para a primeira fase de integração, optou-se por usar um método de crescimento de regiões baseado em critério de homogeneidade de cor-textura. Após estudo de vários sistemas, citam-se Carson et al. (1999); Ozden e Polat (2007); Comaniciu e Meer (2002), optou-se por usar como ponto de partida a técnica de segmentação já bem conhecida chamada JSEG (*JSegmentation*) (Deng e Manjunath, 2001), pois ela apresenta algumas características pretendidas pelo sistema: é totalmente automática, não-supervisionada e baseada em um critério de homogeneidade de cor-textura. Em linhas gerais, o JSEG avalia quantitativamente a homogeneidade local de uma região, criando um mapa 3D onde as regiões são vales e as fronteiras são montanhas. A partir deste mapa 3D (*J-image*), processa-se um algoritmo de crescimento de regiões, e

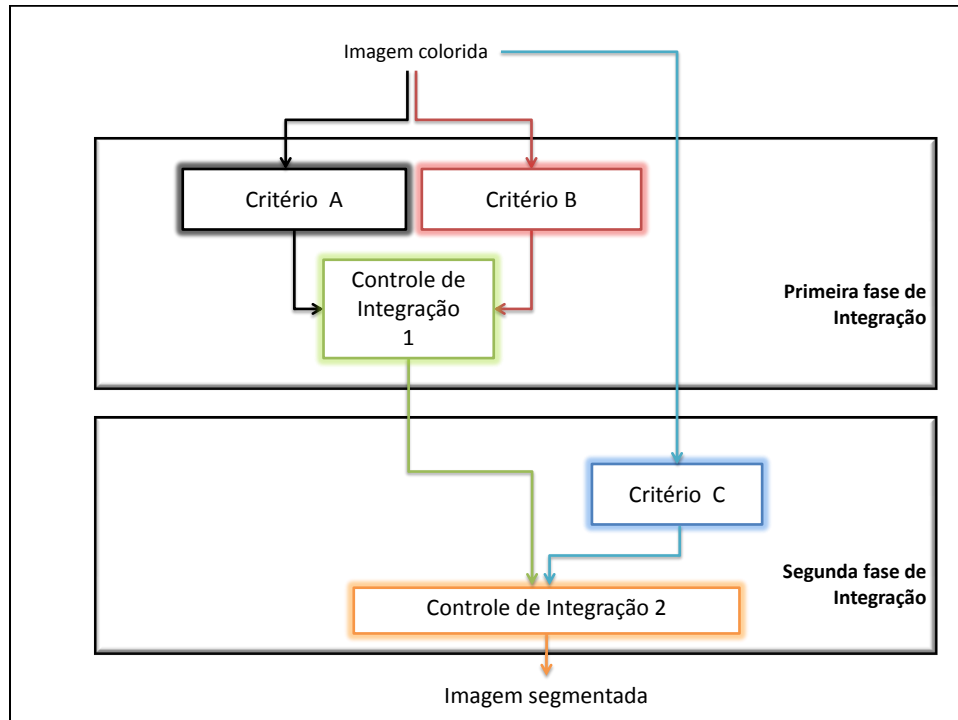


Figura 1.8: Arquitetura genérica da abordagem proposta.

para que o critério seja robusto à granularidade das texturas, o método é multiescalar (ver detalhes sobre o método JSEG no Apêndice A).

A idéia aqui explorada é que é possível melhorar os resultados de segmentação do JSEG integrando outro critério de homogeneidade. É comum usar critérios baseados em forma, modelo, nível de cinza, ou cor-textura (Chandran et al., 2010). Não obstante, o descritor multifractal se encaixa bem na definição de um critério de homogeneidade (Backes, 2010). Trabalhos anteriores (Côco et al., 2009; Côco et al., 2007) motivam a usar uma medida baseada no descritor multifractal como critério de homogeneidade para a melhoria da segmentação de imagens. Para mais detalhes sobre multifractais, ver Apêndice D.

Dois diferentes controles de integração para a primeira fase foram desenvolvidos, um considerando apenas as informações locais e outro baseado em informação global. Esta informação global será baseada em um exame das texturas da imagem de forma global, seguindo o trabalho de Treisman e seus colegas (1977).

Para resolver problemas advindos do, e inerentes ao, método baseado em regiões, foi feita a integração com um método complementar, baseado em descontinuidade. Tal qual o FACADE, teremos um caminho tratando de superfície e outro de borda. A estratégia de integração seguirá a teoria do FACADE, que, através de suas propriedades complementares, o processamento de um caminho guia os sinais do outro caminho. Desta forma, utiliza-se a idéia do “*The Two Streams Hypothesis*”, onde se modela apenas o caminho P. Com

isso, mostra-se, na Figura 1.9, uma implementação da arquitetura genérica da abordagem proposta.

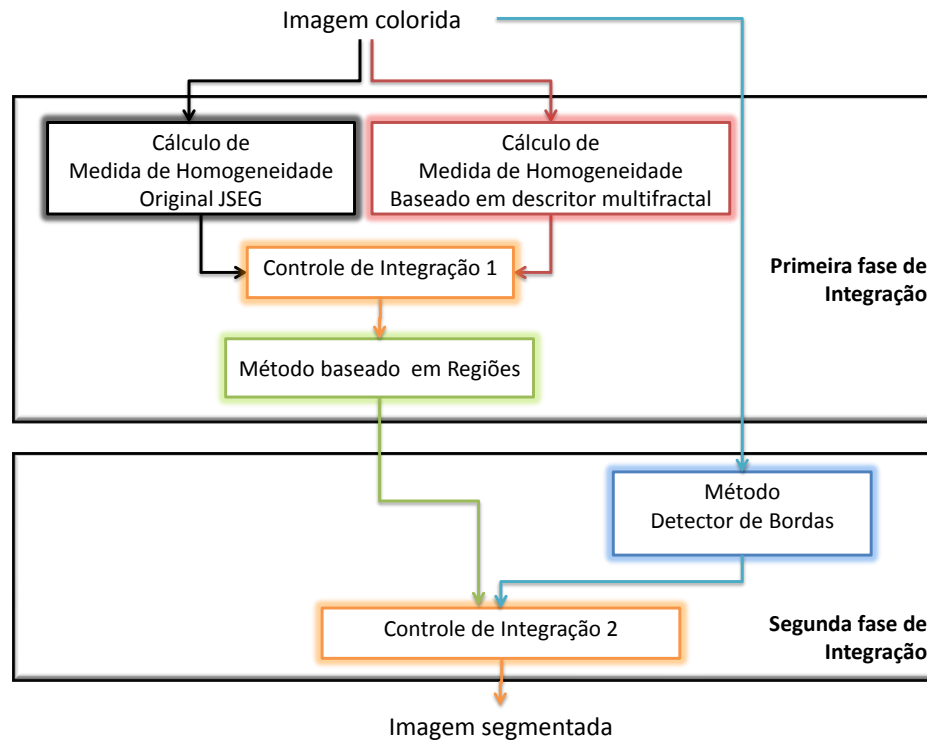


Figura 1.9: Uma instância da arquitetura geral.

1.3 Metodologia de Avaliação dos Resultados

Para a primeira fase de integração, são desenvolvidos quatro diferentes sistemas computacionais. Em cada um deles é usado o sistema JSEG como arcabouço, e as arquiteturas implementadas têm as seguintes características:

1. Arquitetura 1, que usa apenas o descritor multifractal, sem o critério original do JSEG;
2. Arquitetura 2, que é uma extensão da arquitetura 1, baseando-se na integração local entre o descritor multifractal e o critério original do JSEG;
3. Arquitetura 3, que usa apenas o descritor multifractal, mas com forma diferente de combinar os canais do espaço de cores;
4. Arquitetura 4, que é uma extensão da arquitetura 3, onde o descritor multifractal e o critério original do JSEG são integrados por um controle baseado em medida global.

Assim, duas arquiteturas utilizam como critério de homogeneidade apenas o descritor multifractal, e as outras duas integram o descritor multifractal e o critério de homogeneidade original do JSEG. Além disso, implementam-se dois diferentes controles de integração, um baseado em informação local e outro em informação global. Considera-se que a arquitetura zero é o sistema original do JSEG.

A comparação entre métodos é complexa. Há vários requisitos para se efetuar uma comparação: utilização do mesmo banco de dados de imagens, geração dos resultados de segmentação em formatos compatíveis, comparação com o mesmo *benchmark* e cálculo das mesmas métricas. Neste ponto, o que se deseja é comparar os métodos contra a percepção de um ser humano comum, e não um especialista de uma determinada área. Para tanto, utiliza-se o banco de dados BSDS (ver Apêndice C). Neste banco de imagens há uma base de teste composta por 100 imagens coloridas. Para cada imagem do banco de dados, há pelo menos cinco resultados de segmentação de anotadores humanos diferentes, compondo, assim, uma referência para comparação.

Para cada uma das cinco arquiteturas serão feitos os testes usando as imagens da base de testes do banco de dados BSDS, seguindo o modo de utilização descrito em seu manual de uso. A Figura 1.10 ilustra o mecanismo de uso do BSDS. Executa-se cada algoritmo de segmentação usando as 100 imagens da base de testes como entrada, e os resultados, as 100 imagens segmentadas, serão a entrada do processo de cálculo de métricas fornecido pelo próprio BSDS (Martin et al., 2004). Ao final, obtêm-se os valores das métricas *precision* (precisão), *recall* (sensibilidade) e *F-measure* (medida-F). Ver definições e detalhes sobre as métricas no Apêndice B.

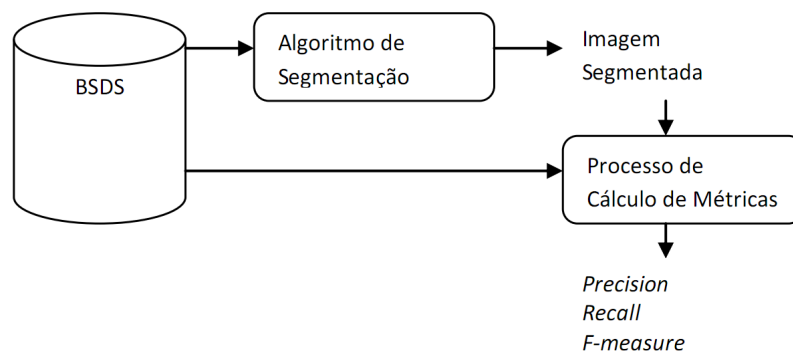


Figura 1.10: Processo de uso do BSDS.

A primeira avaliação é analisar a possível melhoria da segmentação com relação aos métodos individuais - apenas com o JSEG e apenas com o descritor multifractal, tanto qualitativamente quanto quantitativamente. O objetivo é saber se os resultados da arquitetura 2 são melhores que os das arquiteturas 0 e 1, e se a arquitetura 4 é melhor que as arquiteturas

1 e 3. Caso exista evidência de melhoria, então pode-se dizer que a integração destes dois critérios de homogeneidade é melhor que os critérios individuais. A segunda avaliação será analisar se o controle global, da arquitetura 4, foi melhor ou não que o controle local, da arquitetura 2, e com isso verificar qual o melhor controle de integração.

Mesmo usando um banco de dados com as imagens e as referências humanas, ainda há controvérsia nas métricas de qualidade de segmentação. Há artigos que discutem qual seria o melhor método quantitativo, como Wolf e Jolion (Wolf e Jolion, 2006) e Zhang e colegas (Zhang et al., 2008). Nesta fase, as imagens resultantes são todas binárias e, portanto, não há construção de gráficos *precision* \times *recall* nem curva ROC (Fawcett, 2006), pois ter-se-ia apenas um único ponto neste gráficos.

Na segunda fase de integração, utiliza-se a arquitetura de maior métrica da primeira fase de integração, que é integrada com o resultado de um clássico detector de bordas. Para escolher o detector de bordas mais adequado, foram selecionados alguns detectores clássicos e aplica-se o mesmo procedimento do fluxo de trabalho da Figura 1.10. Utiliza-se o método de detecção de bordas cujos resultados obtiveram a maior métrica. Como os resultados dos métodos de detecção de bordas utilizados geram imagens em escala de cinza, será possível plotar os gráficos *precision* \times *recall* para análise.

Foram desenvolvidos dois controles de integração para a segunda fase. No entanto, o segundo é uma extensão do primeiro, onde não houve diferença de concepção, e, sim, apenas uma melhoria. Para ambos os sistemas, utilizou-se o procedimento do BSDS. O objetivo é avaliar se o resultado das integrações promoveu melhoria do resultado com relação aos métodos individuais de entrada, tanto qualitativamente quanto quantitativamente. Nesta segunda fase de integração, como os resultados geram imagens em escala de cinza, será possível plotar gráficos *precision* \times *recall* e curva ROC para análise dos métodos.

1.4 Contribuições

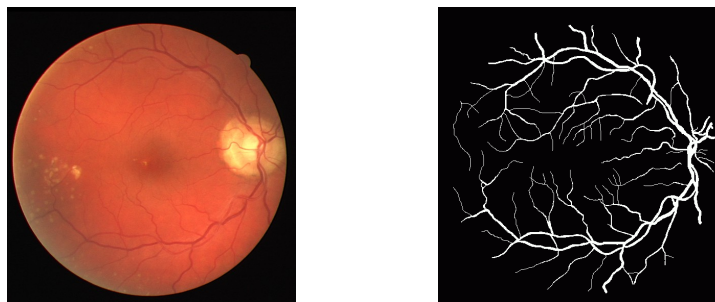
Listam-se as contribuições desta tese:

1. a proposta de um “*framework*”, uma arquitetura genérica, como mostrado na Figura 1.8;
2. uso de um descritor multifractal como critério de homogeneidade de um método de crescimento de regiões;
3. integração do descritor multifractal com o descritor de homogeneidade do JSEG;

4. avaliação quantitativa de métodos detectores de borda segundo métricas calculadas comparada à percepção humana;
5. desenvolvimento de um novo algoritmo de integração de resultados de detecção de bordas com resultados de crescimento de região.

1.5 Limitações

Todas as imagens testadas são imagens com grande variação em cor e textura. Não é objetivo deste trabalho tratar imagens com pouca diversificação de cores, que não apresentem variações de texturas e que possuam regiões muito pequenas ou finas. Por exemplo, há poucas cores em imagens médicas, como as imagens da retina humana encontradas na base de dados DRIVE (Staal et al., 2004). Um exemplo é mostrado na Figura 1.11, onde a imagem à esquerda é a imagem de entrada e a imagem à direita é a imagem segmentada por um especialista. A abordagem deste trabalho identificaria as regiões circulares do globo ocular e da mácula, (que é a região de coloração mais clara na parte direita do globo), e não os vasos sanguíneos como a imagem de referência feita pelo especialista.



(a) imagem de entrada (b) imagem segmentada por especialista

Figura 1.11: Exemplo de imagem de entrada e seu referencial humano da base de dados DRIVE.

Embora a arquitetura proposta tenha como base algumas hipóteses de teorias sobre como se processa o SHV, provar que estas teorias realmente modelam o SHV foge ao escopo desta tese.

Também não é objetivo desta Tese estudar oclusão parcial de objetos, ruído, imagens estéreo, movimento de objetos, a análise de complexidade dos algoritmos ou comparação de tempos computacionais dos sistemas, uma vez que o foco é a proposição de uma metodologia.

Considera-se que a estrutura da segmentação é do tipo “*bottom-up*”. Começa-se com a segmentação e o resultado deste processo é que poderá ser usado posteriormente. Por exemplo, os resultados da segmentação serão a entrada de um sistema de reconhecimento de objetos. Assim, desconsideram-se contribuições “*top-down*”, isto é, não se considera que há camadas superiores à da segmentação que possam fornecer informações de alto nível que auxiliem o processo de segmentação.

1.6 Organização do Texto

Neste capítulo define-se o problema tratado, quais são as dificuldades envolvidas, a motivação, os objetivos, a metodologia, as limitações da proposta, e as contribuições deste trabalho.

O restante do texto foi dividido em quatro capítulos. O segundo capítulo descreve como usar informações suplementares e paralelas em um método de crescimento de regiões, enquanto o capítulo 3 descreve como usar o resultado do crescimento de regiões junto com o resultado de um detector de bordas, isto é, como integrar informações complementares e paralelas. A ordem e divisão dos capítulos 2 e 3 são similares. Inicia-se explicando a fundamentação teórica dos métodos e como foram criadas as arquiteturas baseadas nos mesmos, evidenciando as diferenças entre as várias arquiteturas. Listam-se trabalhos correlatos, apresentam-se os testes e resultados experimentais de forma qualitativa e quantitativa, finalizando com uma discussão destes resultados.

Finalmente, no Capítulo 4 destacam-se as conclusões obtidas deste trabalho e indicam-se as direções futuras para continuação da pesquisa.

Ao final, apresentam-se os apêndices sobre detalhes do método JSEG, as métricas *Precision-Recall* \times ROC, da base de dados BSDS e da teoria de Multifractal.

Capítulo 2

Critério de Homogeneidade Integrando Descritor Multifractal e *J-image*

O que se propõe neste trabalho é uma abordagem não-supervisionada e automática para segmentação de cenas naturais coloridas. A proposta é dividida em duas fases de integração. Na primeira fase, o propósito é usar duas diferentes medidas como critério de homogeneidade no processo de crescimento de regiões. Para atingir tal objetivo, comparam-se os diferentes critérios de homogeneidade individualmente e depois em conjunto. Nas arquiteturas propostas, trabalha-se com a *J-image*, (critério original do JSEG), e o descritor multifractal separadamente, e depois integram-se as duas medidas.

O capítulo inicia descrevendo as características essenciais do JSEG (para detalhes ver o Apêndice A), explica-se como a imagem 3D associada ao descritor multifractal é gerada, e depois detalham-se quatro diferentes arquiteturas propostas, chamadas *Fractal-only* (Komati et al., 2009, 2010b), *Fractal-JSEG* (Komati et al., 2009, 2010b), *I-Frac* (Komati et al., 2010a) e *MM-Frac* (Komati et al., 2011c).

2.1 O método JSEG

Para segmentação de imagens coloridas, um dos métodos mais populares é o método proposto por Deng e Manjunath (2001), o algoritmo JSEG (*J-SEGmentation*). Um dos principais objetivos do JSEG é criar uma abordagem não supervisionada e automática onde não seja necessário estimar um modelo específico de texturas e regiões. A proposta dos autores é avaliar a homogeneidade local de uma região, caracterizada por cor e textura, e assim realizar a segmentação. O JSEG é um método multi-escala de crescimento de regiões, e é muito

usado como padrão de comparação por novas propostas (Haindl e Mikes, 2008; Liu et al., 2007; Celebi et al., 2010).

Na essência, o método JSEG tem dois estágios sequenciais: a quantização de cores e a segmentação espacial. No primeiro estágio, o método trata apenas as cores, independente de sua localização na imagem, gerando uma imagem simplificada que é a entrada para o segundo estágio. O segundo estágio analisa apenas a localização espacial dos elementos, não mais tratando as cores.

O objetivo do primeiro estágio é diminuir a quantidade de cores sem degradar significativamente a qualidade das mesmas. O resultado deste estágio diminui a quantidade de cores, tipicamente entre 10 (dez) a 20 (vinte) cores (Deng et al., 1999a). Cada cor é rotulada como uma classe e, a partir de então, trabalha-se apenas com este mapa-de-classes, e não mais com as cores. Desta forma, a imagem colorida original, que tem três matrizes, é reduzida a apenas uma matriz, que segue como entrada para o estágio de segmentação espacial.

No estágio de segmentação espacial calcula-se a medida J , que é o critério que quantifica a distribuição de classes. Essencialmente, mede a distância entre diferentes classes, dividindo-a pela distância entre os membros de cada classe, conceito similar ao discriminante linear multi-classes de Fisher. O valor J é calculado para cada pixel, usando uma janela local, gerando assim uma imagem, a *J-image*. A multi-escala é feita alterando-se o tamanho da janela local, criando uma *J-image* para cada escala. Na *J-image*, quanto maior o valor de J local maior é a probabilidade do pixel estar próximo a uma fronteira entre regiões. A *J-image* é como um mapa 3-D contendo vales e montanhas, que representam o centro e as fronteiras das regiões, respectivamente. As características da *J-image* permitem utilizar um método de crescimento de regiões para a segmentação da imagem. Finalmente, para diminuir o problema de supersegmentação, as regiões são mescladas (*region merge*), com base em semelhança de cores. Após a junção de regiões, a segmentação final é obtida.

Mostram-se as imagens intermediárias do algoritmo na Figura 2.1. Tem-se em (a) a imagem de entrada, que é a imagem 101085 do BSDS, porém com as dimensões alteradas para que ficasse quadrada, (esta alteração só foi feita para melhorar a disposição das imagens no texto). Na sequência da Figura 2.1, em (b) apresenta-se a imagem após o processo de quantização de cores, em (c) a *J-image* na escala 3, em (d) a imagem segmentada associada a (c), em (e) a *J-image* na escala 2, em (f) a imagem segmentada associada a (e), em (g) tem-se a referência humana do BSDS e em (h) apresenta-se a segmentação final após o *region merge* das escalas 2 e 3. As imagens (d), (f) e (h) estão sobrepostas sobre a imagem de entrada. O número das escalas, 2 e 3, referem-se aos tamanhos das janelas locais da Tabela A.1. Nesta figura, observa-se que:

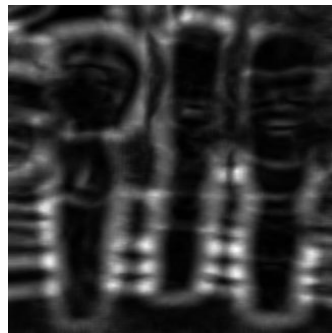
- a imagem quantizada realmente possui uma quantidade menor de cores. Neste caso,



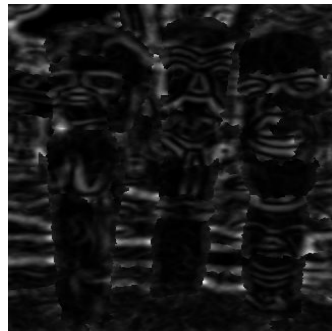
(a) 101085



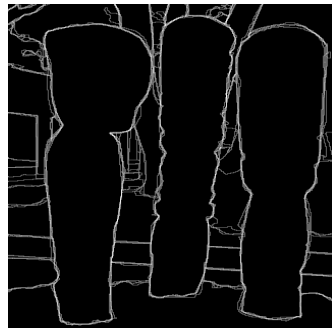
(b) após quantização

(c) J -image na escala 3

(d) segmentação da escala 3

(e) J -image na escala 2

(f) segmentação da escala 2



(g) referência humana



(h) segmentação final

Figura 2.1: Resultados intermediários do método JSEG para a imagem 101085 do BSDS.

após a quantização, há apenas 8 (oito) cores;

- a imagem *J-image* da escala 2 apresenta mais detalhes do que a imagem *J-image* da escala 3. Este fato é esperado, pois a escala 2 possui uma janela local menor, e o tamanho da janela é que determina o tamanho de regiões que podem ser detectadas;
- o processo de *region merge* (resultado mostrado na Figura 2.1h) diminuiu a quantidade de regiões da escala 2, e apresenta maior quantidade de regiões que o resultado da escala 3. Este processo seleciona as regiões, não criando novas regiões ou mesmo alterando a localização das fronteiras entre as regiões.

O método JSEG, assim como os métodos de crescimento de regiões, apresenta resultados com fronteiras finas e conectadas, e é estável com relação ao ruído. Alguns bons resultados de segmentação do JSEG são mostrados na Figura 2.2a, b e c (os números identificam as imagens dentro da base de dados BSDS). Uma constatação importante é que melhores resultados do JSEG surgem quando os elementos das cenas naturais possuem cores bem distintas.



Figura 2.2: Resultados de segmentação do método JSEG.

No entanto, em alguns casos, os resultados de segmentação do JSEG apresentam o problema de supersegmentação, que consiste em caracterizar um número de regiões muito grande, separando regiões que um ser humano não separaria. Observa-se que o método

JSEG apresenta muita segmentação quando há textura fina, como pode ser visto na Figura 2.2d, onde cada totem está sendo subdividido em várias partes, ou como na Figura 2.2e, onde a vegetação está sendo segmentada em várias partes. O método também segmenta cada janela da Figura 2.2f, numa imagem onde os anotadores humanos indicam o prédio com contorno único.

Assim, acredita-se que é possível melhorar os seus resultados de segmentação adotando um critério mais adequado para distinguir informações de características intra e inter-regiões. Considera-se que operadores de detecção de bordas não são compatíveis com tal tarefa, pois são muito sensíveis a quaisquer bordas e não seriam bons critérios de homogeneidade de regiões.

A Figura 2.3 mostra, em sua parte (a), um mosaico com três texturas, em (b) o resultado da imagem após o filtro de Marr-Hildreth, em (c) o resultado após filtro de Canny e em (d) o resultado invertido da imagem do descritor multifractal para tamanho de janela 3×3 . Observe como os resultados dos detectores de borda se comportam: o filtro de Marr-Hildreth é tão sensível que responde a qualquer alteração na intensidade dos valores de cinza. Mesmo o filtro de Canny, sendo menos sensível do que o de Marr-Hildreth, indica cada reta do entrelaçado da região superior direita. Já a imagem associado ao descritor multifractal enfatiza as fronteiras entre as regiões e responde menos forte para as bordas dentro das regiões. Se aumentarmos o tamanho da janela, a resposta às informações dentro das regiões se torna mais fraca ainda, pois o tamanho da janela está associado a granularidade da textura a ser reconhecida.

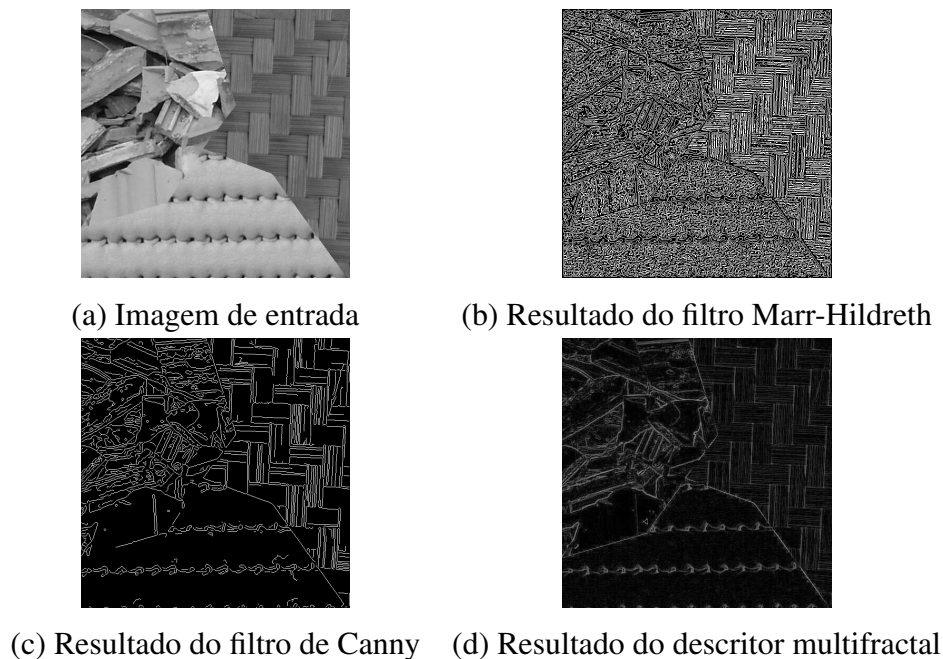


Figura 2.3: Aplicação de diferentes filtros em um mosaico de texturas.

Trabalhos anteriores (Côco et al., 2009; Côco et al., 2007) motivam o uso de uma medida baseada no descritor multifractal como um critério de homogeneidade de regiões baseado no padrão cor-textura. Conforme Côco (2007), o descritor multifractal não se resume apenas a um simples detector de bordas. Backes (2010) afirma que a dimensão fractal atua como uma medida de homogeneidade da textura, tornando possível a sua quantificação e a sua comparação com outras texturas. O Apêndice D contém informações e referências sobre o assunto.

2.2 A imagem do descritor multifractal

Desde o trabalho de Pentland (1984), a idéia de visualizar os valores da dimensão fractal como uma imagem já existia. Porém, ele não gerava um valor por pixel, mas sim por bloco de pixels. Véhel e Mignot também usam esta ideia de visualização em seu artigo de 1994 (Véhel e Mignot, 1994). Os resultados destes trabalhos é que motivam, nesta Tese, tratar o cálculo da dimensão fractal como se fosse um filtro espacial de processamento de imagens de detecção de fronteiras entre regiões homogêneas. Ao invés de usar a equação D.6, usou-se a equação

$$(q - 1)D_q(i, j) = \tau(q) = \frac{\ln [\sum_{i,j} \mu_r(i, j)^q]}{\ln \left(\frac{M}{r}\right)}, \quad (2.1)$$

A alteração feita, a retirada do limite, significa que ao invés de fazer a relação linear no gráfico \ln/\ln usou-se o valor computado diretamente como indicação de borda no mapa 3D. Segundo Véhel e Mignot (1994) isso é possível, pois a idéia principal não é calcular os valores “reais” do espectro multifractal, mas sim, informar quando se está ou não perto de uma fronteira entre regiões. Em outros termos, o objetivo não é o cálculo exato dos valores, mas sim evidenciar diferenças entre os valores estimados. Logo, o interesse é que os valores calculados representem apenas a indicação de fronteiras. Com isso, tem-se uma diminuição significativa do tempo de computação, pois não será necessário efetuar a regressão linear em cada pixel da imagem.

Em geral, calcula-se o valor da dimensão para vários valores de q . Porém, Teng e colegas (Teng et al., 2010) mostram que os valores para $q > 2$ podem ser representados pelo resultado correspondente ao valor $q = 2$ e valores $q < -1$ podem ser representados pelo resultado de $q = -1$. Isto é, apenas os resultados de q no conjunto $-1, 0, 1, 2$ contém informações relevantes, e fora dessa faixa os valores são redundantes. Theiler (1990) afirma que o método de *Box-Counting* para o cálculo da dimensão multifractal, quando $q < 1$, não é adequado, e que

se tem melhor estimativa para $q > 1$. (Teng et al., 2010) ainda afirma que para $q = 2$, regiões regulares apresentarão alto valor para D_2 (equação 2.1 para $q = 2$) e, conseqüentemente, valores baixos para regiões não-regulares.

Assim, ao se gerar uma imagem 3D com a informação do descritor multifractal local, as fronteiras serão vales estreitos e profundos confinados entre paredes abruptas, quase verticais. Na Figura 2.4, apresenta-se em (a) a imagem 101085 da base de dados do BSDS, com as dimensões alteradas para que a imagem fique quadrada. As imagens (b), (e) e (f) apresentam as imagens dos descritores multifractais respectivamente para as janelas 9×9 , 17×17 e 33×33 . Nas imagens (a), (b), (e) e (f) está representada a linha 200 (duzentos) na cor branca, para ressaltar qual foi a linha utilizada para mostrar o gráfico 2D, respectivamente em (c), (d), (g) e (h).

É interessante observar que a imagem do descritor multifractal segue o que foi explicado, isto é, regiões regulares apresentarão valores elevados (pixels mais claros) e, conseqüentemente, regiões não-regulares apresentarão valores baixos (pixels mais escuros). As fronteiras entre regiões são regiões não-regulares, e, portanto, são apresentadas como um contorno escuro. À medida que a janela aumenta, as bordas da imagem do descritor multifractal também vão se tornando mais grossas.

Os valores J , no método JSEG, formam uma imagem 3D, a *J-image*, onde os vales representam áreas de regiões homogêneas e as montanhas representam as fronteiras entre regiões. Portanto, tem-se uma representação inversa nas duas imagens, a *J-image* e a imagem gerada pelo descritor multifractal. Efetua-se uma inversão dos valores nas imagens do descritor multifractal, para que ambas as representações tenham a mesma semântica. Também é feita uma normalização dos valores para que ambas as representações tenham os mesmos limites de valores, sendo que o valor do *J-value* varia entre 0 e 2. A Figura 2.5 mostra, na coluna da esquerda, as imagens dos descritores fractais normalizados para as janelas 9×9 (parte a), 17×17 (parte c) e 33×33 (parte e). Nota-se que estas imagens são mais parecidas com as imagens do *J-image* apresentadas nas Figuras 2.1c e 2.1e. Na coluna direita da Figura 2.5 apresentam-se os gráficos 2D da linha 200 da imagem à esquerda.

Para facilitar a comparação da localização das fronteiras entre regiões da referência humana com relação à imagem do descritor multifractal, mostram-se dois gráficos na Figura 2.6. O primeiro gráfico, em (b), apresenta duas curvas: a curva em azul é a linha 200 da imagem de entrada 101085 e a curva em vermelho é a linha 200 da imagem de referência humana, mostrada em (a). Embora exista uma indefinição logo no início do gráfico, perto da origem, a curva em vermelho mostra basicamente 6 (seis) picos, que são exatamente as fronteiras dos três totens. Pela curva em azul, são os pontos onde há uma singularidade no gráfico como um degrau. No segundo gráfico, em (c), a curva em vermelho é a mesma de (b)

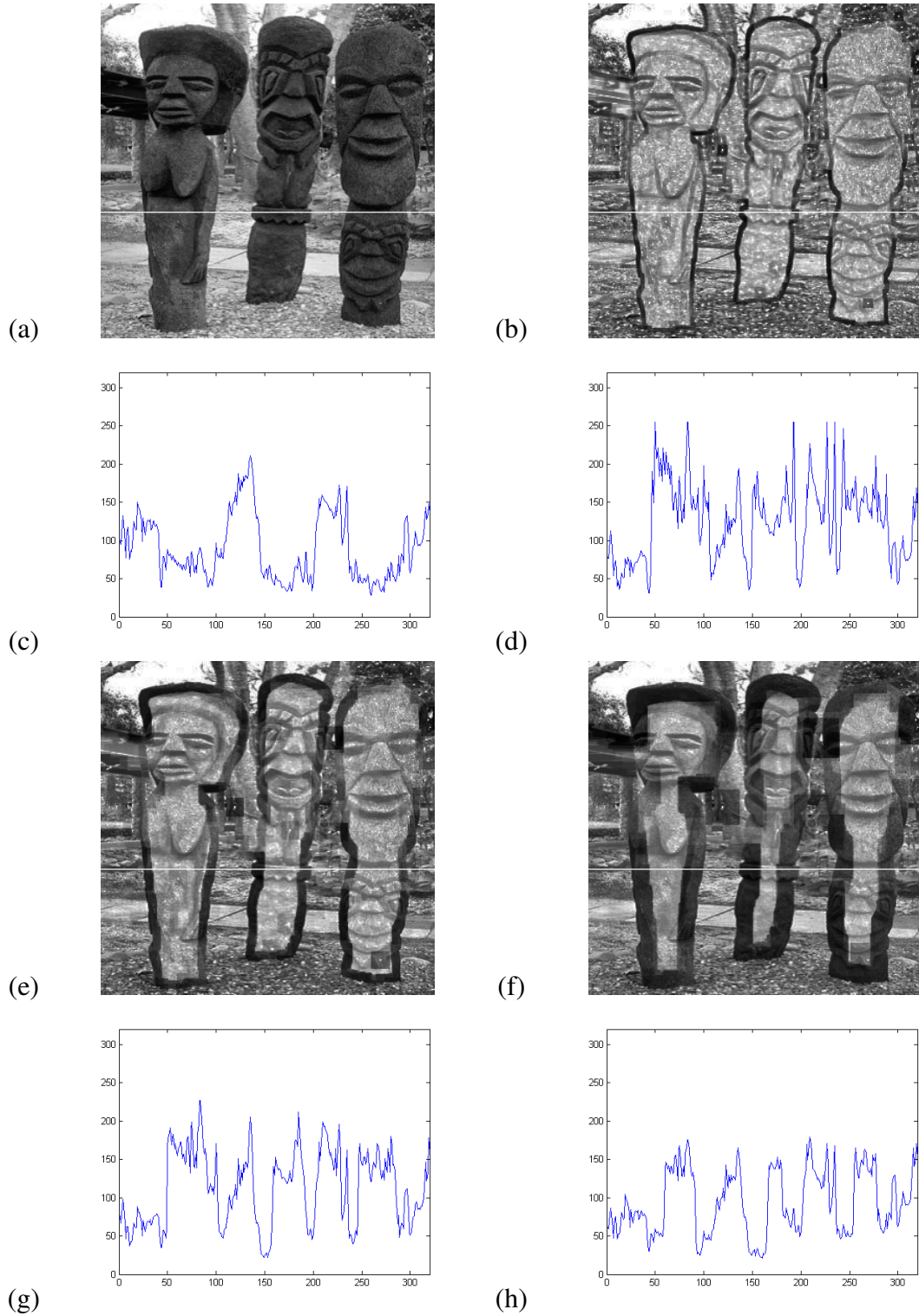
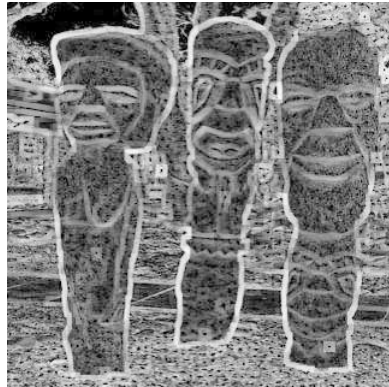
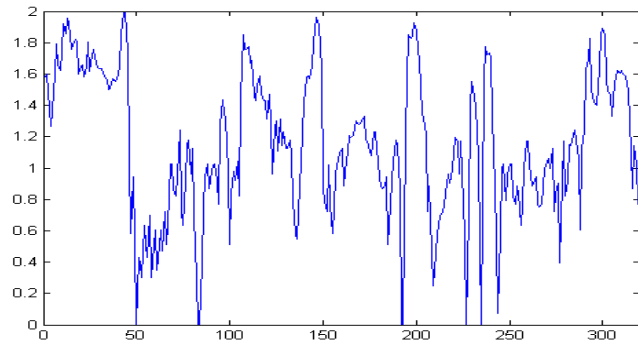
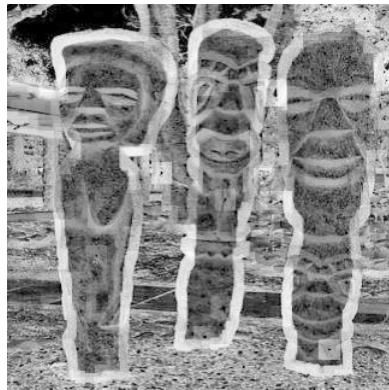
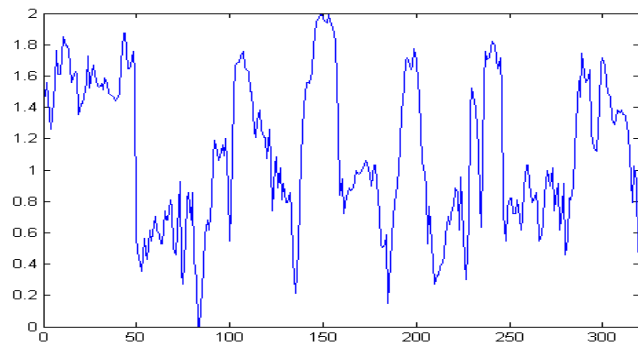


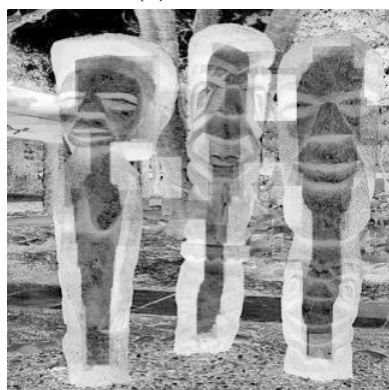
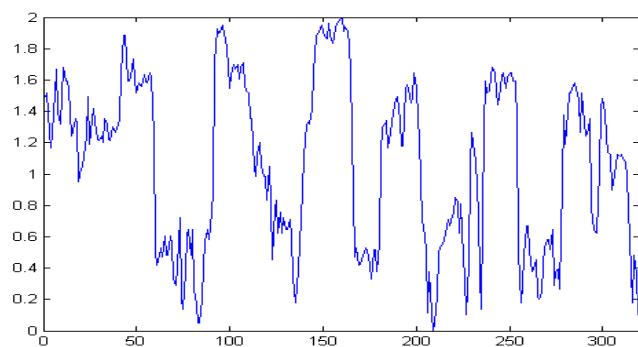
Figura 2.4: (a) Imagem 101085 da base de dados do BSDS; (b) Imagem do descritor fractal para janela de tamanho 9×9 ; (c) gráfico 2D da linha 200 da imagem (a); (d) gráfico em 2D da linha 200 da imagem (b); (e) Imagem do descritor fractal para janela de tamanho 17×17 ; (f) Imagem do descritor fractal para janela de tamanho 33×33 ; (g) gráfico 2D da linha 200 da imagem (e); (h) gráfico em 2D da linha 200 da imagem (f).

(a) 9×9 

(b)

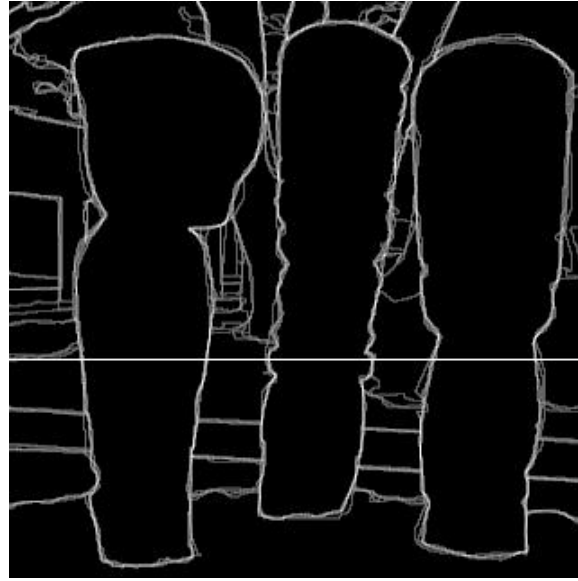
(c) 17×17 

(d)

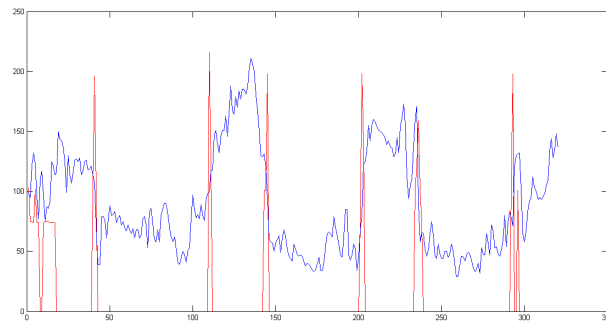
(e) 33×33 

(f)

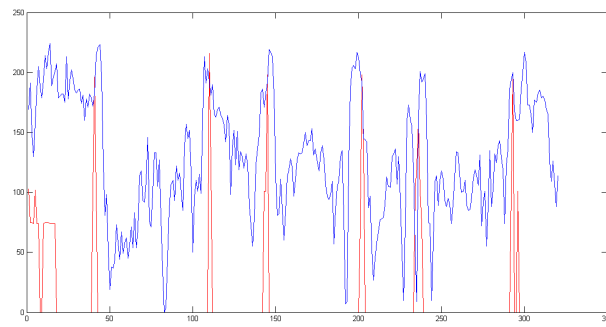
Figura 2.5: À esquerda apresentam-se as imagens dos descritores fractais normalizados, (o tamanho da janela associada se encontra abaixo de cada imagem). À direita, apresentam-se os gráficos 2D da linha 200 da imagem à esquerda.



(a)



(b)



(c)

Figura 2.6: (a) Referência Humana de 101085. (b) Gráfico 2D com as curvas da linha 200 da imagem em escala de cinza 101085 em azul (c) Gráfico 2D com as curvas da linha 200 da imagem do descritor multifractal com janela 9×9 em azul. A curva em vermelho de (b) e (c) corresponde à linha 200 da imagem da referência humana.

e a curva em azul é a linha 200 da imagem do descritor multifractal com janela 9×9 , já com os valores invertidos, mas não normalizados. Os picos das curva em azul abrangem os picos da curva em vermelho. O fato da localização não ser exata será resolvido pelo algoritmo de crescimento de regiões, que irá aglutinando os vales até chegar às montanhas.

2.3 Trabalhos Relacionados

Há poucos trabalhos propondo melhorias no método JSEG, dentre os quais dois são aqui apresentados. Nos trabalhos de Chang e colegas (Chang et al., 2007a,b) se propõe um método de melhoria de contraste, chamado IC-JSEG. Este método altera o espaço de cores e altera o contraste da imagem, gerando um mapa de contraste. Aplicam-se dois filtros nesse mapa, um para retirada de ruído e outro para enfatizar as bordas. Testes em três imagens médicas e em três imagens de cenas naturais são realizados, e os autores mostram que houve melhoria no problema de supersegmentação do JSEG.

No trabalho de Wang e colegas (Wang et al., 2006), os autores propõem uma abordagem para solucionar o problema de supersegmentação de áreas com gradação suave de cores que o JSEG apresenta. Para tanto, embute-se uma medida de operadores direcionais de Frei e Chung-Ching (1977) para identificar a similaridade de cores. Testes com imagens do BSDS são realizados, mas os resultados quantitativos não são baseadas nas métricas padrão atuais do BSDS, impossibilitando a comparação quantitativa com este método.

A dimensão multifractal é amplamente utilizada para segmentação de imagens ou texturas. A começar pelo trabalho de Pentland (1984), cuja técnica consiste em dividir a imagem em blocos e, para cada bloco, calcular a dimensão fractal. Depois, constrói-se um histograma com os valores da dimensão fractal, que é então dividido em seus vales ou mínimos locais. Cada divisão do histograma representa uma área homogênea da imagem. Há três exemplos de imagens de cenas naturais em escala de cinza, e Pentland relata outros testes mostrando bons resultados, porém lamenta que não há como comparar seus resultados com outras técnicas.

Véhel possui vários artigos nesta área, escritos com colegas variados, onde se destaca seu trabalho de 1994 (Véhel e Mignot, 1994). Neste trabalho, ele demonstra que é possível usar características multifractais de cada pixel da imagem para solucionar o problema de segmentação. Estes valores são as entradas de um modelo de otimização Bayesiana, que classificará os pontos em pixels de borda ou não. Os experimentos, em imagens em escala de cinza, mostraram que esta abordagem é melhor, quando comparada com o detector de bordas Canny. Já Chaudhuri e Sarkar (1995) utilizam um vetor de características, tendo-se

dimensão fractal local e dimensão multifractal, entre outros valores, como entrada para o método de *clustering*, o *K-means*. Bons resultados foram alcançados ao segmentar mosaicos de texturas em escala de cinza.

Maeda e colegas (Maeda et al., 2000), apresentam um algoritmo para estimar a dimensão fractal local (DFL) de imagens, com base no *Blanket Method* (Peleg et al., 1984). Depois, unem características de diferença de intensidade de cores, gradiente de cores e DFL de cada pixel e aplicam um procedimento chamado *Fuzzy Region-Growing*. O teste é feito em apenas uma imagem de cena natural colorida. Kasparis e colegas (Kasparis et al., 2001) usam um vetor de características composto por medidas fractais, e um algoritmo iterativo baseado em *K-means* para a segmentação. O algoritmo de agrupamento inicia em uma janela local grande, que a cada iteração é reduzida. Comparações com características baseadas em energia são feitas em testes com imagens de mosaicos de Brodatz.

Nunes e Conci (2003) propuseram o uso de fractais para segmentação de texturas em imagens multi-bandas. Além da geração da dimensão multifractal local de cada banda, é feito um exame do inter-relacionamento entre a representação da imagem em cada banda. Após uma fase de treinamento supervisionado, é feita a classificação automática dos valores. Os testes de segmentação foram feitos com mosaicos de texturas naturais coloridas e imagens reais de satélites, e através deles os autores mostram a eficiência do método.

Xia e colegas (Xia et al., 2006) propuseram o LMME (*Local Morphology-based Multifractal Estimation*), que é um novo algoritmo para cálculo da dimensão multifractal usando técnicas de morfologia matemática. As características são computadas através de uma janela deslizante de 11×11 pixels, centrada no ponto. A segmentação é feita com base no algoritmo de agrupamento *fuzzy C-means*. Ao final, o trabalho afirma, através de resultados de testes, que o LMME diferencia melhor texturas que duas outras técnicas de estimativas de dimensão multifractal, MDBC (Houqiang et al., 2001) e MRDBC (Du e Yeo, 2002) em mosaicos de texturas e cenas naturais.

Côco, Salles e Sarcinelli-Filho (Côco et al., 2009) apresentaram um novo critério topográfico para o método TICA (*Topographic Independent Component Analysis*), uma combinação da energia mais o descritor LMME. Os vetores de características são submetidos ao método *K-means*. Resultados em mosaicos de texturas mostram que a nova abordagem com o descritor multifractal é melhor que a tradicional, que usa apenas energia.

Portanto, uma abordagem comumente utilizada é criar um vetor de características, que depois é analisado por um método de agrupamento, como o *K-means*. Trabalhos anteriores, inclusive, já haviam utilizado descritores multifractais como um elemento num vetor de características. Nesta Tese, diferentemente, será usado um descritor multifractal local para criar uma representação de uma imagem 3D com vales e montanhas, indicando o quanto

o pixel está ou não numa região de borda. Assim, ao invés de se criar um vetor de características utilizado como entrada para um algoritmo de agrupamento, propõe-se uma nova abordagem, que consiste em integrar diferentes medidas em um único mapa, diminuindo o esforço do algoritmo de agrupamento.

2.4 Arquiteturas Implementadas

A partir do JSEG foram desenvolvidos quatro diferentes sistemas, já mencionados na seção 1.3, sumarizadas na Tabela 2.1.

Tabela 2.1: Características das Arquiteturas Implementadas.

#	Nome	Combinação dos Canais de Cores	Descritor Multifractal	<i>J-image</i>	Controle
1	Fractal-only	Função de máximo	Mapa de Classes	Não	
2	Fractal-JSEG	Função de máximo	Mapa de Classes	Sim	Local
3	I-Frac	Funções de máximo e média	Imagem de Entrada	Não	
4	MM-Frac	Funções de máximo e média	Imagem de Entrada	Sim	Global

Deve ser enfatizado que duas arquiteturas utilizam como critério de homogeneidade de regiões apenas o descritor multifractal (arquiteturas 1 e 3), e as outras duas integram o descritor multifractal com o critério de homogeneidade original do JSEG (arquiteturas 2 e 4). Foram implementados dois diferentes tipos de controle de integração, um baseado em informação local e outro em informação global (no caso das arquiteturas 2 e 4). As arquiteturas 1 e 2 usam o mapa de classes, isto é, o resultado após o processo de quantização de cores como entrada para o processo de cálculo das imagens do descritor multifractal, enquanto as arquiteturas 3 e 4 usam a imagem colorida original. As arquiteturas 1 e 2 usam a função de máximo para combinar as imagens do descritor multifractal de cada canal de cor, e as arquiteturas 3 e 4 combinam os canais de cores através de funções de máximo e média.

Durante os testes preliminares foram utilizados os espaços de cores: o RGB, LUV e o Lab. Os melhores resultados foi no espaço de cores LUV, que já era utilizado pelo JSEG.

2.4.1 Fractal-only

A arquitetura chamada *Fractal-only* usa somente as imagens de dimensão multifractal como critério de homogeneidade do algoritmo de crescimento de regiões, conforme pode ser visto na Figura 2.7. Os elementos que estão sombreados mostram as inclusões feitas, em relação ao sistema original JSEG. É importante salientar que se mantiveram os processos de quantização de cores e crescimento de regiões do método original. Conforme explicado, o cálculo da dimensão multifractal local é executado pelo método DBC, e o tamanho do cubo utilizado é o mesmo que o da janela local para calcular o *J-image*.

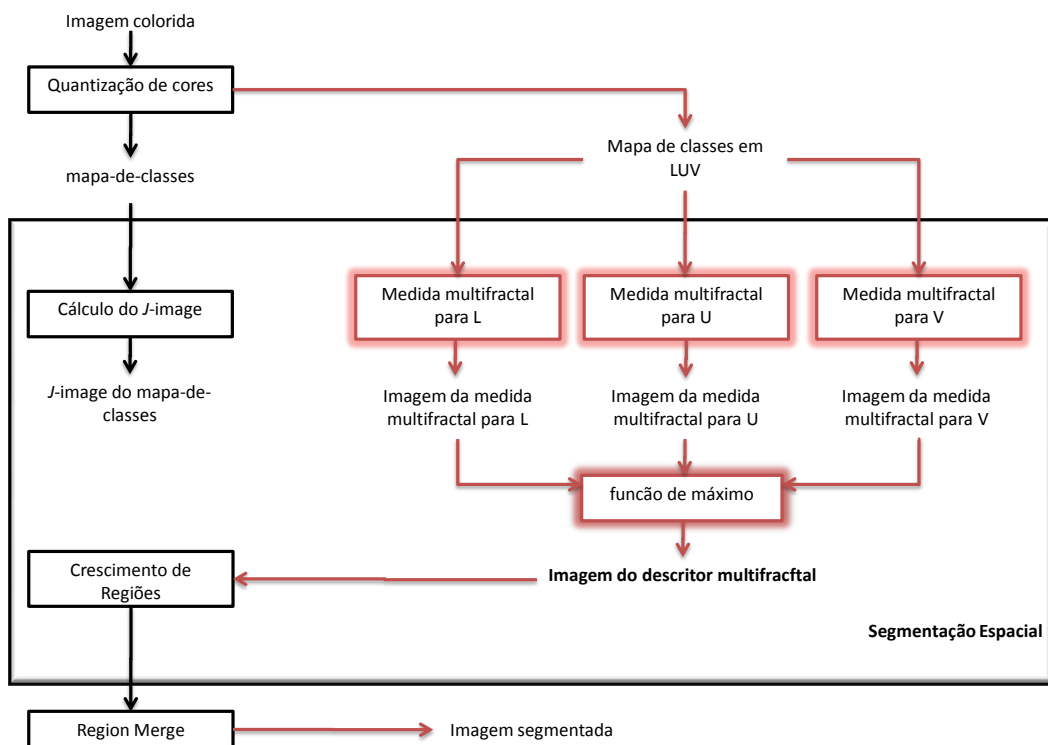


Figura 2.7: Arquitetura do *Fractal-only*.

Os resultados produzidos pelo método JSEG são baseados principalmente no mapa de classes produzido pela primeira etapa, onde se tem o processo de quantização de cores. O mapa de classes é formado por números na faixa de valores entre 1 (um) e C , onde C é o número de classes. Por exemplo, se a quantização reduz a quantidade de cores para dez cores, então o mapa de classes possui valores entre 1 (um) e 10 (dez).

Entretanto, observa-se que a medida J é definida sobre a variância deste mapa de classes, que descreve a distribuição das classes, mas acaba desconsiderando a informação de cor.

Assim, para o cálculo do descritor multifractal, ao invés de se usar o mapa de classes, usa-se o valor da cor média de cada classe. Continuou-se a usar o espaço de cores perceptualmente uniforme LUV, também conhecido como CIELUV, que já era usado pelo método JSEG. Por exemplo, ao invés de tratar a informação do pixel como classe de valor 1, tratou-se a informação do pixel como cor azul (com 3 valores, um para luminosidade e dois relativos à cor). Com isso, obtém-se maior sensibilidade às variações de cor que no método original.

Os cálculos das três imagens de dimensão multifractal dos componentes do espaço de cores são executados separadamente. Todos os mapas 3D são combinados por uma função de máximo. Esta abordagem, de tratar cada canal do espaço de cores de forma independente, provém de Cheng e colegas (Cheng et al., 2001). Esta idéia também é usada em Conci e Proenca (1997), porém no espaço de cores RGB.

Usando a função de máximo, cada pixel do mapa resultante tem valor maior ou igual aos valores dos mapas de entrada, na mesma posição do pixel. Se o descritor multifractal de um pixel no componente U é maior que a dimensão do mesmo pixel no componente L, significa que há uma informação de borda mais perceptível na cor (U) que na luminosidade (em L). Associando sempre o valor máximo de cada pixel, aumenta-se a quantidade de montanhas, isto é, define-se melhor a localização das fronteiras entre regiões.

2.4.2 Fractal-JSEG

O objetivo desta arquitetura é analisar se o resultado dos critérios de homogeneidade, (*J-image* e o descritor multifractal), combinados apresentam resultados melhores ou piores que os critérios isoladamente.

A Figura 2.8 mostra a arquitetura do sistema Fractal-JSEG. Os elementos que estão sombreados mostram as inclusões feitas, em relação ao sistema original JSEG.

A diferença entre o Fractal-JSEG e o Fractal-only foi o uso do mapa 3D *J-image* como entrada da função de máximo, além dos mapas 3D dos descritores multifractais. O cálculo das três imagens do descritor multifractal dos componentes do espaço de cores e do *J-image* são executados separadamente.

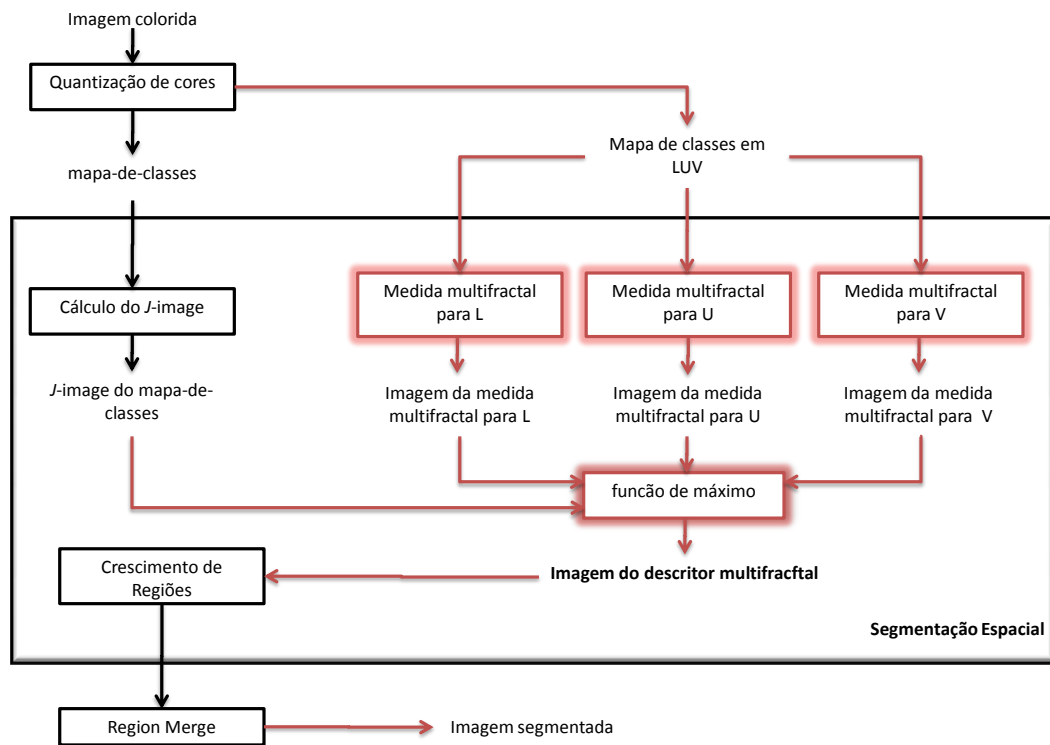


Figura 2.8: Arquitetura do Fractal-JSEG.

2.4.3 I-Frac

Nesta terceira arquitetura, mostrada na Figura 2.9, há três pontos que a diferenciam do Fractal-only:

1. uso da imagem original para o cálculo da imagem da dimensão multifractal. A dimensão fractal fornece uma informação quantitativa da rugosidade da superfície. Ao se quantizar a imagem, isto é, diminuir a quantidade de cores da imagem, altera-se a informação original. Essa diminuição da informação afeta o cálculo de rugosidade. Assim, ao usar a imagem original como entrada para o cálculo da imagem do descritor multifractal tem-se uma informação mais precisa sobre a rugosidade da superfície, tal qual no trabalho de Kasparis e colegas (Kasparis et al., 2001);
2. combinação das imagens de dimensão multifractal. No Fractal-only todos os mapas 3D são combinados através de uma função de máximo. Isto é, supõe-se que cada componente do LUV tem a mesma importância para a definição de bordas. Testes preliminares mostraram que os componentes de cores, U e V, têm informações similares,

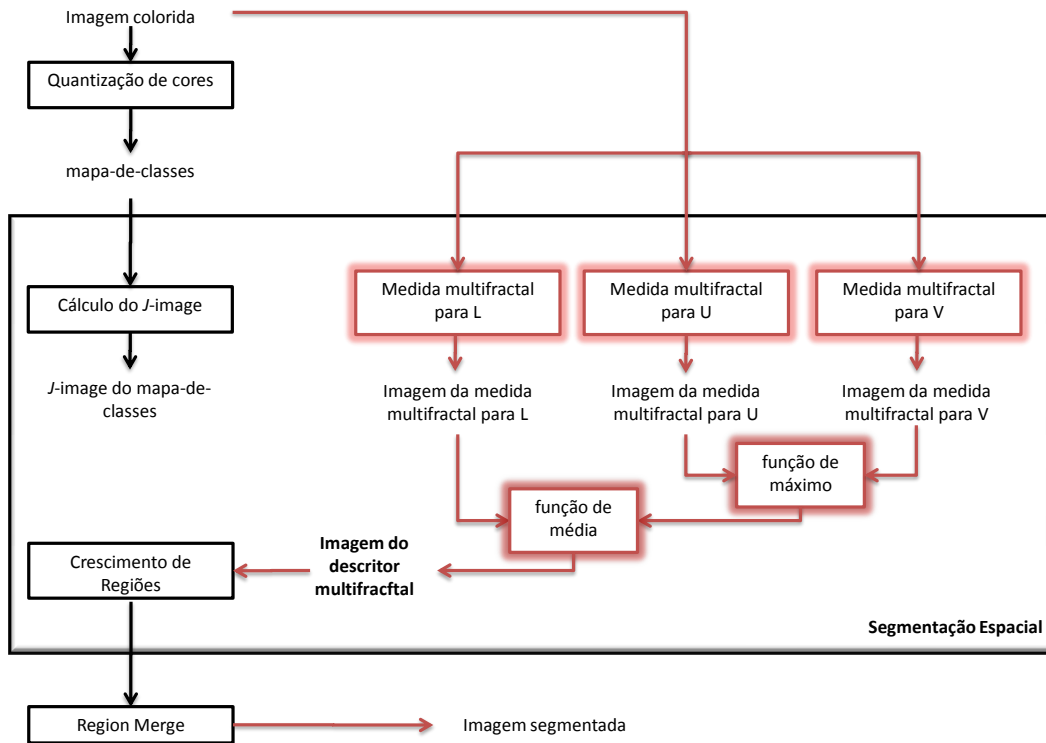


Figura 2.9: Arquitetura do I-Frac.

e que uma combinação diferente pode fornecer uma imagem de bordas mais descritiva. Nestes testes, a melhor combinação foi o uso da função de máximo entre os componentes de cores e o uso da função de média entre os resultados da luminância e da combinação de cores;

- alteração da janela local. Reduziu-se à metade os tamanhos originais das janelas locais do JSEG, com isso reduzindo o tamanho dos cubos. Além disso, não há a necessidade de se diminuir a amostragem. A janela local utilizada no JSEG possui um esquema de subamostragem, isto é, diminui a quantidade de informações da janela (veja Tabela A.1 do Apêndice A). No artigo original que propõe o método do JSEG (Deng e Manjunath, 2001) esta subamostragem é feita por motivos computacionais. Entretanto, para os computadores atuais pode-se relaxar esta restrição, de forma que nesta arquitetura não há mais esta subamostragem. Logo, todos os pixels da janela local (cubo) são considerados. Assim, a tabela de escalas adotadas é a Tabela 2.2.

Tabela 2.2: Tamanho das janelas para cada escala do I-Frac.

escala	janela (pixels)	amostragem (1/pixels)	tamanho da região (pixels)	tamanho mínimo da semente (pixels)
1	5×5	1/(1×1)	32×32	16
2	9×9	1/(1×1)	64×64	32
3	17×17	1/(1×1)	128×128	128
4	33×33	1/(1×1)	256×256	512

2.4.4 MM-Frac

A arquitetura MM-Frac utiliza o mapa 3D *J-image*, além dos mapas 3D dos descritores multifractais, para compor a entrada para o processo de crescimento de regiões. O cálculo das três imagens do descritor multifractal dos componentes do espaço de cores e do *J-image* são executados separadamente. A Figura 2.10 mostra a arquitetura do sistema MM-Frac, onde todos os retângulos representam os processos que recebem imagens como entrada e retornam uma imagem como saída. O processo de estimativa de α é representado como uma elipse, pois é o único processo que recebe uma imagem como entrada e retorna um número real como saída.

Diferente da arquitetura do Fractal-JSEG, o controle de mistura dos mapas agora se baseia em uma propriedade estatística global da imagem: o valor α , que é a declividade da curva do espectro de potência da imagem de entrada. Esta medida se mostrou muito eficiente na classificação de imagens de cenas naturais em diferentes categorias semânticas (objetos simples, cômodos, cenas panorâmicas, dentre outros), conforme trabalho de Torralba e Oliva (2003), e também na segmentação de imagens, como no trabalho de Côco, Salles e Sarcinelli-Filho (Côco et al., 2009). Eles observaram que a declividade do perfil da curva dada pelo espectro de potência das imagens ($\log(\text{magnitude}) \times \log(\text{frequência})$) recai, na média, na forma $1/f^\alpha$ com $\alpha \sim 2$.

A curva é dada pelos valores decrescentes do espectro de energia, da frequência mais baixa para a mais alta. A Figura 2.11 exemplifica o espectro de potência em 3D, onde o perfil da curva é enfatizado em vermelho. A Figura 2.12 mostra o perfil (em vermelho), no gráfico 2D, e a interpolação linear (linha preta pontilhada). No exemplo, o valor estimado é dado por $-\alpha = -2,31$ (ou $\alpha = +2,31$).

Estatísticas de imagens naturais parecem seguir um padrão de regularidade. Pentland (1984) mostrou que superfícies naturais fractais (como montanhas e florestas) produzem uma imagem com espectro de energia na forma $1/f^\alpha$, onde α está relacionado com a dimensão

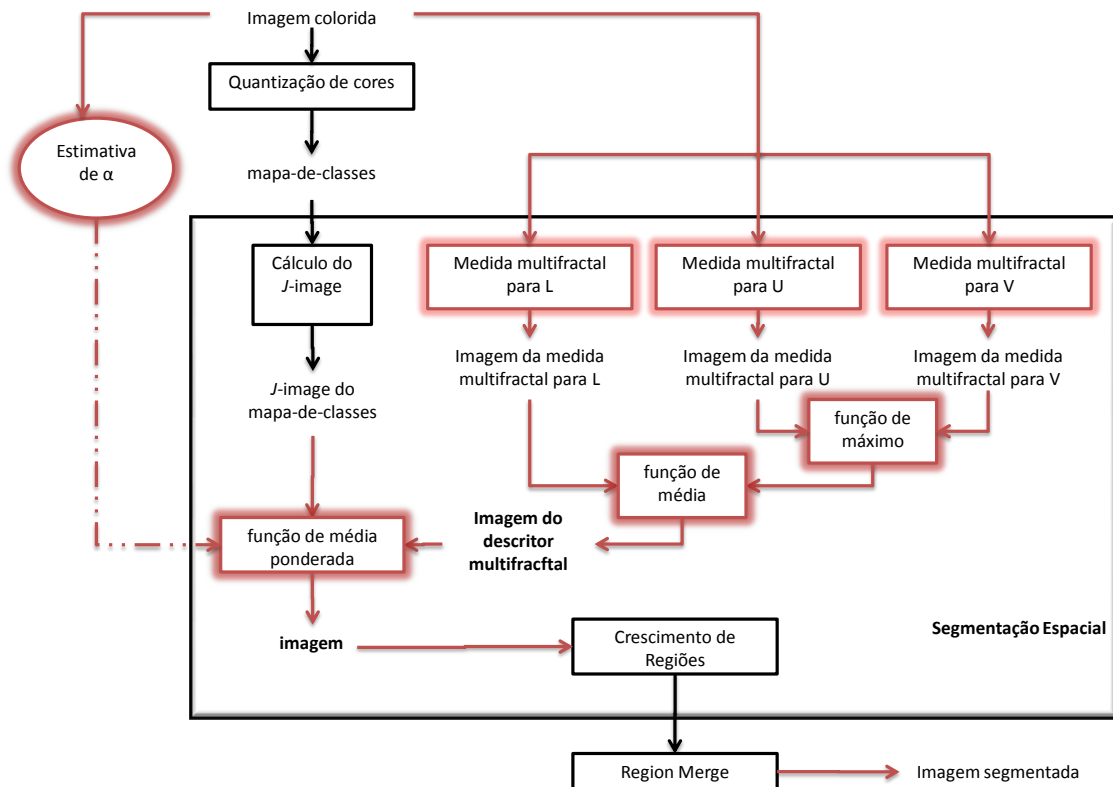


Figura 2.10: Arquitetura de MM-Frac.

fractal de superfícies 3D (isto é, a sua rugosidade). Características de declividade podem ser agrupadas em duas famílias: declividade suave ($\alpha \sim 1$), para cenas rugosas e com objetos detalhados, e declividade acentuada ($\alpha \sim 3$), para cenas com objetos de pouca rugosidade (mais lisos). Logo, quanto mais suave é a declividade, mais rugosa é a imagem ou a imagem possui mais informação de alta frequência.

Alguns exemplos de imagens com muita rugosidade são apresentados na coluna esquerda da Figura 2.13, enquanto que na sua coluna direita são apresentados os respectivos gráficos da interpolação de α . Nestes gráficos mostra-se, também, a reta com declividade -2, apenas para facilitar a comparação visual. A estimativa de α foi feita apenas no componente de luminância, por isso as imagens estão em escala de cinza. Na Figura 2.14 são mostrados alguns exemplos de imagens com $\alpha \sim 3$. Da mesma forma que na figura anterior, na coluna esquerda estão as imagens de entrada e na coluna direita estão os gráficos associados a seus respectivos α . Na figura com baixo valor de α tem-se muitas rugosidade, como a areia e a vegetação com galhos. Já na outra figura, as imagens apresentam regiões extensas com as mesmas cores ou de pouca rugosidade, como nuvens.

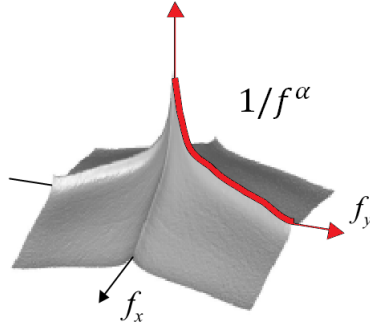


Figura 2.11: Espectro de potência de uma imagem em 3D.

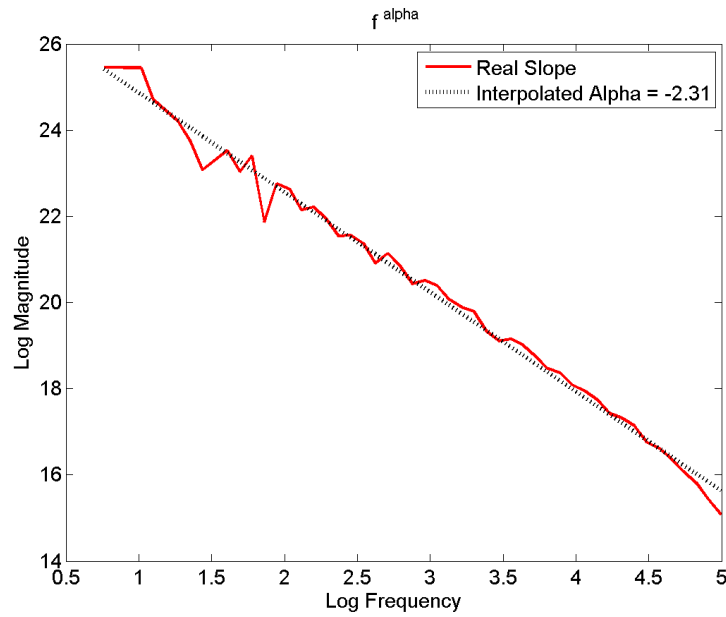


Figura 2.12: Espectro de potência de uma imagem em 2D.

Para integrar as duas medidas, utilizou-se uma média ponderada baseada em α . Quanto menor o valor de α , maior o peso da medida multifractal. Assim, cada pixel no mapa 3D final é dado por

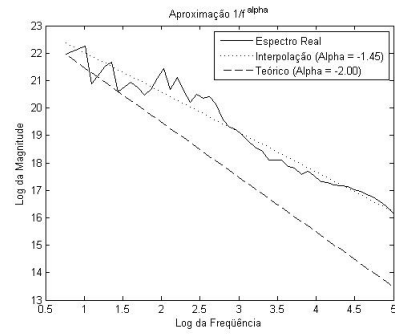
$$map_{ij} = J-value \times \alpha_{norm} + (1 - \alpha_{norm}) \times valor-Fractal, \quad (2.2)$$

onde $\alpha_{norm} = (\alpha - 1, 0)/2, 0$. Esta normalização foi feita considerando a faixa de valores [1,3] citado no trabalho de Pentland (1984), de tal forma que α_{norm} esteja na faixa [0,1].

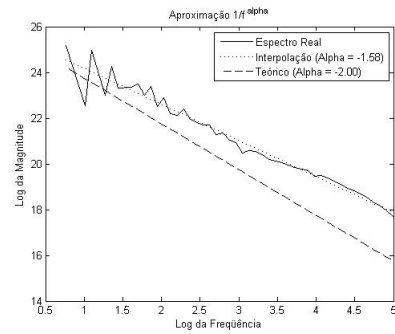
Um valor baixo de α , significa que a imagem apresenta muita rugosidade, e o peso da parcela relativa ao mapa do descritor multifractal é privilegiado com relação à parcela do *J-image*. Para valores altos de α , os pesos se invertem.



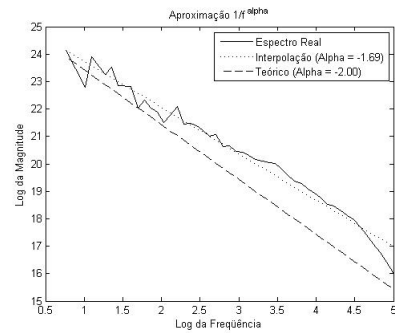
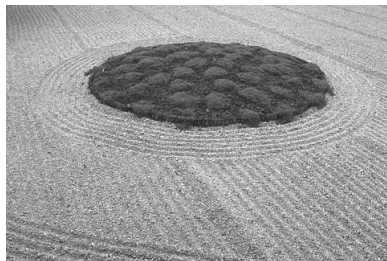
196073

 $\alpha=1,45$ 

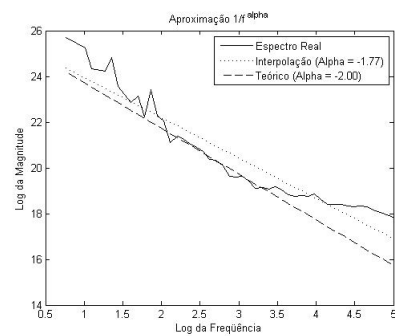
175032

 $\alpha=1,58$ 

69040

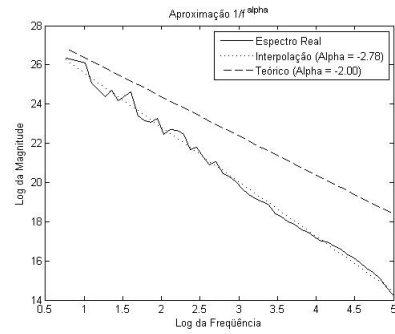
 $\alpha=1,69$ 

86016

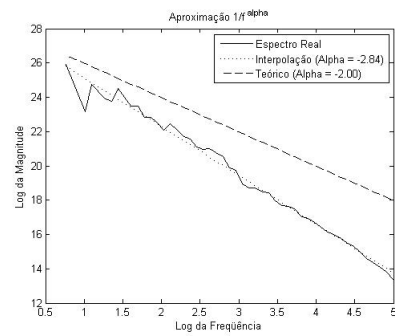
 $\alpha=1,77$ Figura 2.13: Exemplos de imagens com valores de α próximos de 1.



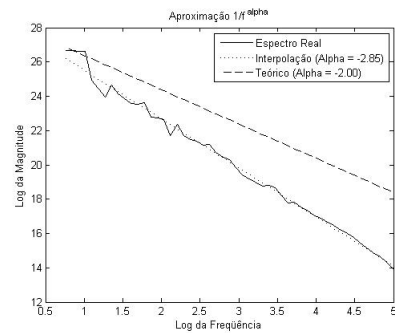
106024

 $\alpha=2,78$ 

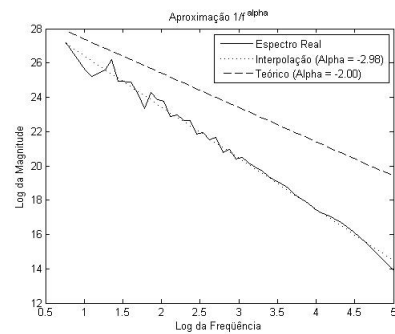
3096

 $\alpha=2,84$ 

45096

 $\alpha=2,85$ 

189080

 $\alpha=2,98$ Figura 2.14: Exemplos de imagens com valores de α próximos de 3.

2.5 Resultados Experimentais e Discussão

Nesta seção são mostrados e analisados os resultados de cada uma das quatro arquiteturas descritas na seção anterior, de forma qualitativa e quantitativa. Os experimentos não incluíram qualquer tipo de calibração de parâmetros para alguma imagem específica. Isto é, todos os parâmetros foram configurados para valores padrão, e não houve modificação destes valores durante as baterias de testes. O limiar para quantização de cores e a quantidade de escalas são calculados de forma automática. Para o limiar do *region merge* foi utilizado o valor padrão do JSEG, o qual é 0,4.

2.5.1 Comparação entre JSEG e Fractal-only

Os resultados e a análise serão apresentados na seguinte sequência:

1. Figura 2.15, mostrando os resultados onde o método Fractal-only obteve métricas de *F-measure* maiores que o método JSEG;
2. Figura 2.16, mostrando os resultados onde o método Fractal-only obteve métricas de *F-measure* menores que o método JSEG;
3. Figura 2.17, mostrando os resultados onde o método JSEG obteve métricas de *F-measure* maiores que o método Fractal-only;
4. Figura 2.18, mostrando os resultados onde o método JSEG obteve métricas de *F-measure* menores que o método Fractal-only.

A ordem em todas as figuras é a mesma. Há quatro colunas, sendo que a primeira mostra a imagem original da base de dados de testes (BSDS), seguida pela sua imagem de referência composta pelas anotações humanas (*benchmark*), seguida pelos resultados das segmentações usando o método JSEG e os resultados das segmentações pelo método Fractal-only.

Analisando os resultados das segmentações do método Fractal-only, nos quais o valor de *F-measure* foram maiores que os valores de *F-measure* dos resultados do JSEG, verificou-se que há uma característica comum nestas imagens: texturas complexas são identificadas como uma região única. Destacam-se alguns exemplos desta característica na Figura 2.15. Observa-se, na primeira linha, que o ser humano não segmenta cada janela, e sim o contorno do prédio como um todo, e que a segmentação resultante do Fractal-only é mais parecida com a percepção humana que o resultado do JSEG. Na segunda linha da figura, nota-se que no resultado do JSEG, os totens foram divididos em mais sub-regiões que no resultado do

Fractal-only. Assim, nestas imagens, usando apenas o critério de homogeneidade baseado no descritor multifractal, diminui-se a quantidade de regiões.

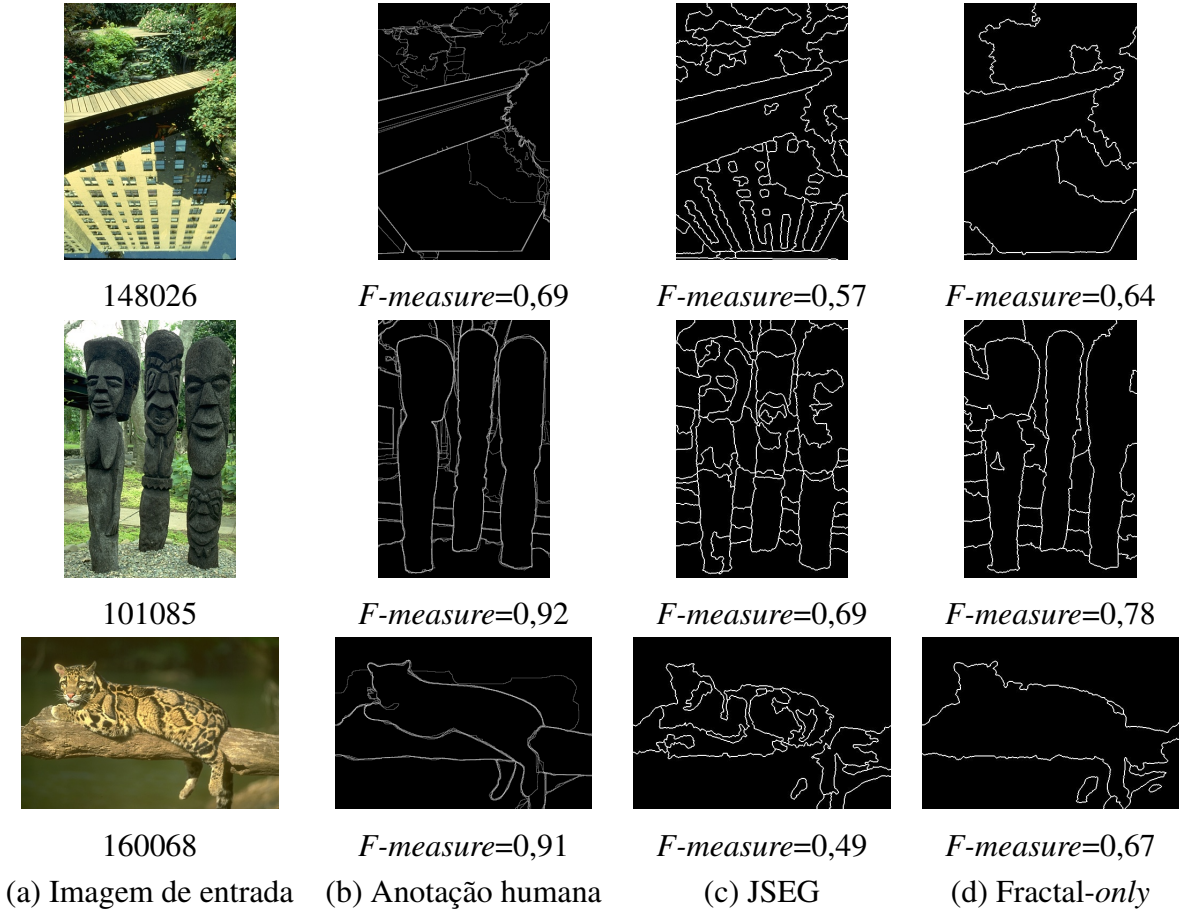


Figura 2.15: Exemplos característicos com texturas complexas nos quais o Fractal-only apresentou valores de $F\text{-measure}$ maiores que o JSEG.

A terceira linha da Figura 2.15 apresenta métrica melhor que o JSEG, e realiza uma tarefa extremamente difícil de agregar a textura complexa da onça numa única região. No entanto, a generalização foi tal que abarcou, inclusive, o galho da árvore.

Embora a generalização do descritor multifractal tenha fornecido bons resultados em algumas imagens, em outras os resultados pioraram, conforme pode ser visto na Figura 2.16. A primeira linha da figura mostra o pior resultado, que não apresenta sequer um pixel de borda. Considerando-se que o resultado da segmentação será a entrada de um processo de alto nível (por exemplo, reconhecimento de objetos), seria necessário que existisse pelo menos algum indício de que há algum elemento central (embora supersegmentação não seja desejável, ela é melhor que nenhuma segmentação). Este resultado provém principalmente do processo de quantização que diminui para apenas duas cores diferentes. Outro exemplo deste problema aparece na figura do bode nas pedras, na segunda linha da figura, onde o

resultado do *Fractal-only* não consegue isolar o contorno do bode das rochas.

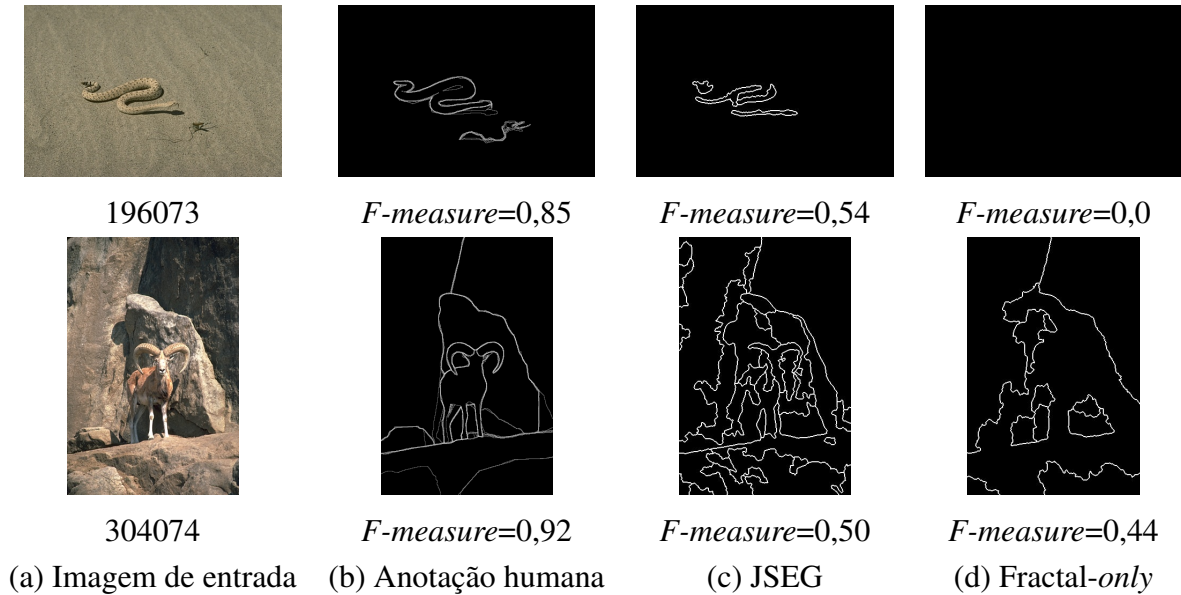


Figura 2.16: Exemplos característicos com similaridade de cor-textura de diferentes componentes, nos quais o *Fractal-only* apresentou valores de $F\text{-measure}$ menores que o JSEG.

Verificando todos os piores resultados do *Fractal-only*, notou-se que há também uma característica comum. As imagens apresentam cores do elemento central e cores do fundo muito similares. No geral, o método *Fractal-only* se mostra instável, apresentando alguns poucos resultados surpreendentes, mas com a desvantagem de não conseguir distinguir elementos de cores muito similares.

Já se pode observar que o método JSEG tende a supersegmentar as imagens. Há imagens em que esta característica é bem-vinda, como no caso da primeira linha da Figura 2.17. A folhagem da parte inferior esquerda da imagem é bem detalhada pela anotação humana: o ser humano percebeu as folhas e não uma região de folhagem, e com isso, o resultado do JSEG apresenta métrica maior que o método *Fractal-only*. Da mesma forma, na segunda linha a imagem de referência humana foi bem detalhista quanto às vestimentas dos homens, e obteve maior $F\text{-measure}$ que o *Fractal-only*.

Além das imagens com grande complexidade de textura nas quais a percepção humana compreende como uma única região, como mostrado na Figura 2.15, onde o JSEG apresentou desvantagens, mostra-se ainda mais um caso em que ele apresenta métrica menor que o *Fractal-only*. No caso da Figura 2.18, imagem de entrada não apresenta tanta complexidade de textura, mas há regiões de texturas com pouca rugosidade ao fundo, que o JSEG segmenta, embora o mesmo não seja segmentado na referência humana. Por sua vez, o *Fractal-only* não segmenta o fundo, mas também não consegue uma boa definição no contorno do lobo.

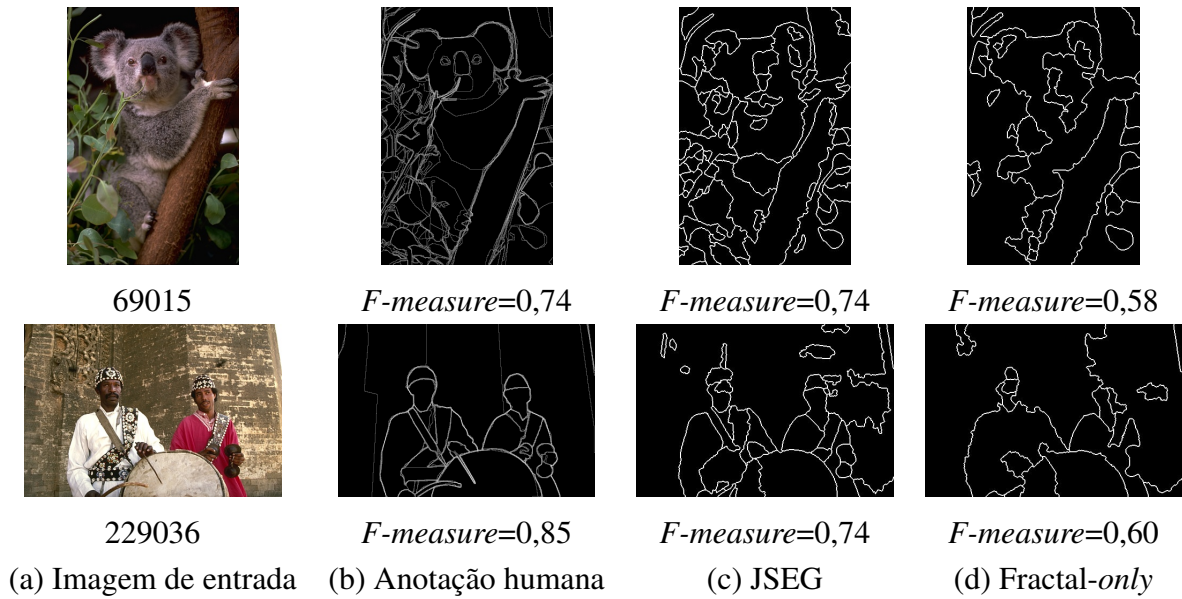


Figura 2.17: Exemplos característicos onde as anotações humanas foram detalhadas e nos quais o JSEG apresentou valores de $F\text{-measure}$ maiores que o Fractal-only.

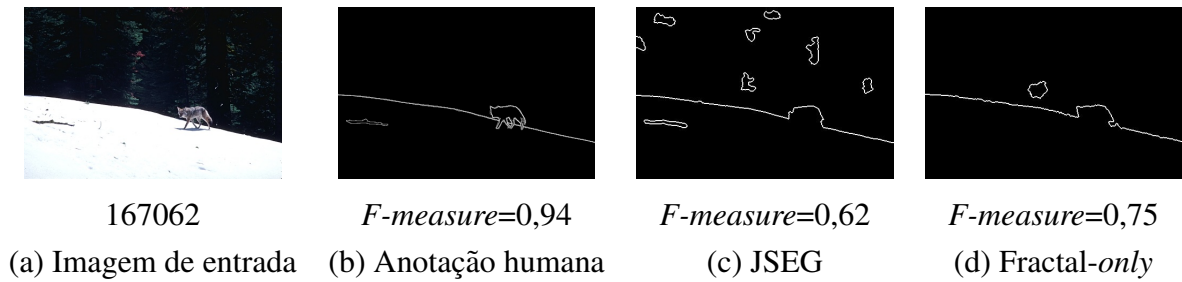


Figura 2.18: Imagens associadas a imagem 167062.

2.5.2 Comparação Fractal-JSEG x (JSEG e Fractal-only)

Conforme já comentado, dependendo das características dos vários componentes das imagens, um dos critérios de homogeneidade pode ter desempenho melhor ou pior que o outro. Para analisar o seu funcionamento em conjunto, uma das contribuições desta Tese, foi a proposta da arquitetura Fractal-JSEG. Exemplos de imagens nos quais os resultados do Fractal-JSEG foram melhores, em termos de $F\text{-measure}$, que os resultados do JSEG e do Fractal-only são apresentados na Figura 2.19.

O resultado do Fractal-JSEG, em geral, mostra uma mistura entre os dois critérios, como na primeira linha da Figura 2.19. O resultado do Fractal-JSEG não apresenta as regiões ao fundo, como no resultado do Fractal-only, e consegue definir melhor o contorno do lobo, como no resultado do JSEG. O mesmo ocorre com relação ao canguru da segunda linha e aos totens da terceira linha: os componentes apresentam os contornos tal qual os resultados

do JSEG, mas contêm menos regiões, como nos resultados do Fractal-only.

As imagens resultantes das segmentações de todos os métodos são do tipo *hard edge maps*, isto é, são imagens binárias onde a informação é simplesmente se um pixel é ou não pixel de fronteira entre regiões. Por esta razão, não há um gráfico tradicional de *precision* \times *recall* para cada imagem, onde cada ponto no gráfico representaria um limiar de binarização numa imagem do tipo *soft-edge-map*.

Os gráficos apresentados nas Figuras 2.20a, 2.20b e 2.20c são histogramas. A leitura se dá da seguinte forma: na Figura 2.20b, cada ponto representa a quantidade de imagens que tiveram o *recall* na faixa de valores do eixo *x*. Logo, pode-se ver que o JSEG apresentou 27 imagens (de um total de 100 imagens de teste da base BSDS) com *recall* = 0,6. As leituras para os histogramas das métricas *precision* e *F-measure* também ocorrem da mesma forma.

Analisando os gráficos de *precision* e *recall*, vemos que o JSEG apresenta melhor *recall*, mas tem os piores resultados de *precision*. O método Fractal-only apresenta melhor resultado em termos de *precision*, mas falha em termos de *recall*. O método Fractal-JSEG apresenta resultado entre JSEG e Fractal-only nas métricas de *recall* e *precision*. Considerando que existe um conflito entre os valores de *precision-recall*, isto é, a melhora de uma métrica, em geral, tem como consequência a piora da outra métrica (Gordon e Kochen, 1989), é normal que as curvas para *F-measure* para todos os métodos fiquem similares, como ocorre no gráfico da Figura 2.20c.

Quanto maior for o máximo valor da curva, e quanto mais à direita for esse máximo, melhor é o desempenho do algoritmo. No gráfico de *F-measure*, é fácil observar que o método Fractal-only apresenta um desempenho pior que os outros dois métodos, e que quase não há diferença entre as curvas dos métodos Fractal-JSEG e JSEG.

Considerando todas as 100 imagens de testes da base BSDS, o Fractal-JSEG apresentou maior valor de *F-measure* em 57, o Fractal-only foi melhor em 17 e o JSEG em 26.

É importante lembrar que a combinação (função de máximo) não é uma função linear, e nem o método de crescimento de regiões é linear. Assim, é possível que se tenha resultados com *F-measure* com valores menores que ambas as entradas (foram 7 casos), assim como é possível se ter resultados com valor de *F-measure* maiores que ambas as entradas (num total de 10 casos).

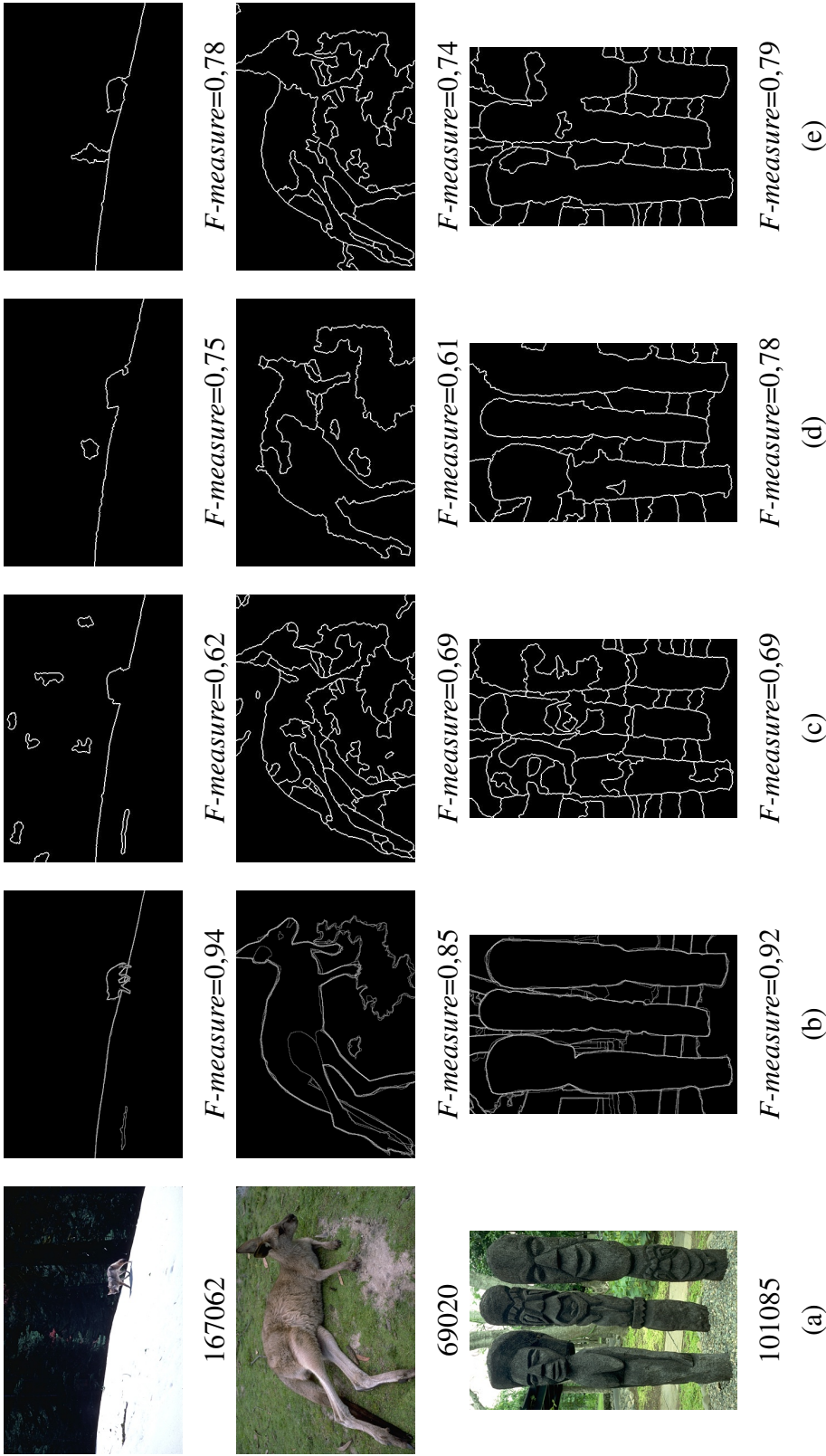
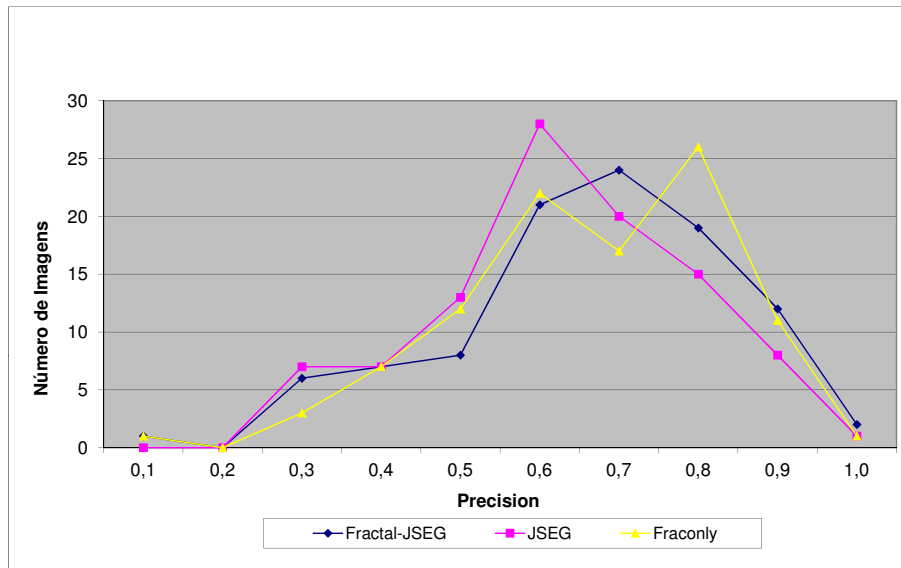
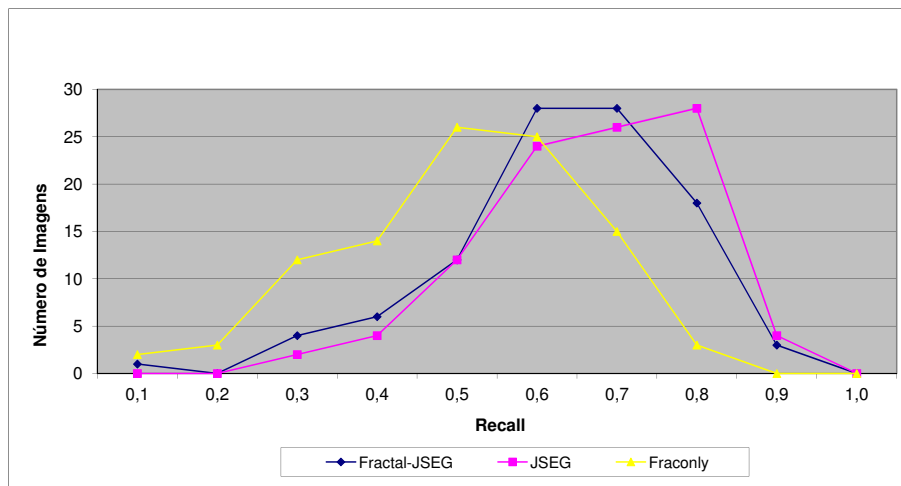
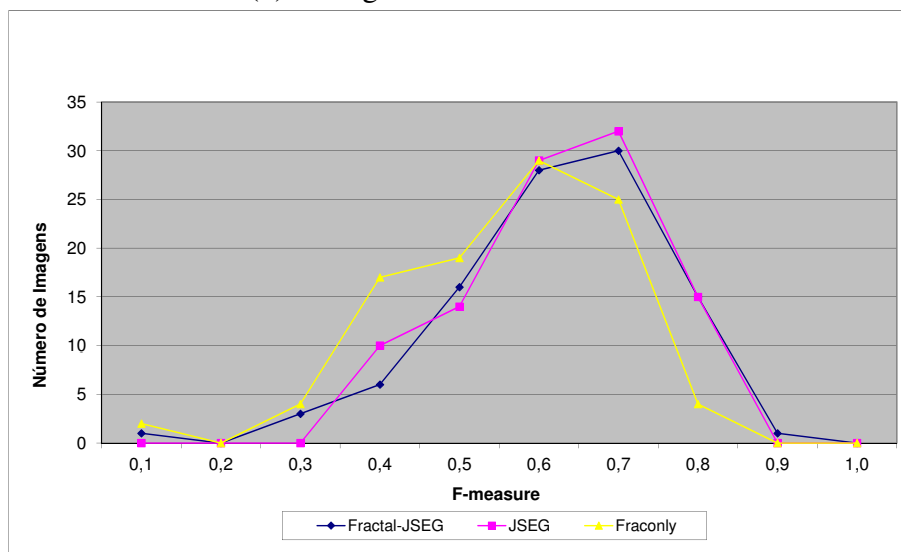


Figura 2.19: Exemplos nos quais o Fractal-JSEG melhorou a métrica *F-measure* com relação aos métodos JSEG e Fractal-only. (a) Imagem de entrada (b) Anotação humana (c) resultado do JSEG (d) resultado do Fractal-only (e) resultado do Fractal-JSEG

(a) Histograma da métrica *precision*.(b) Histograma da métrica *recall*.(c) Histograma da métrica *F-measure*.Figura 2.20: Histogramas das métricas dos métodos JSEG \times Fractal-only \times Fractal-JSEG.

2.5.3 Comparação I-Frac x Fractal-only

O método I-Frac foi proposto devido ao problema do método Fractal-only não distinguir componentes das imagens com cores similares, como visto na Figura 2.16. As mesmas imagens de entrada da Figura 2.16 são repetidas na Figura 2.21(a), em (b) estão a anotação humana associada, em (c) o resultado do Fractal-only e em (d) o resultado do I-Frac.

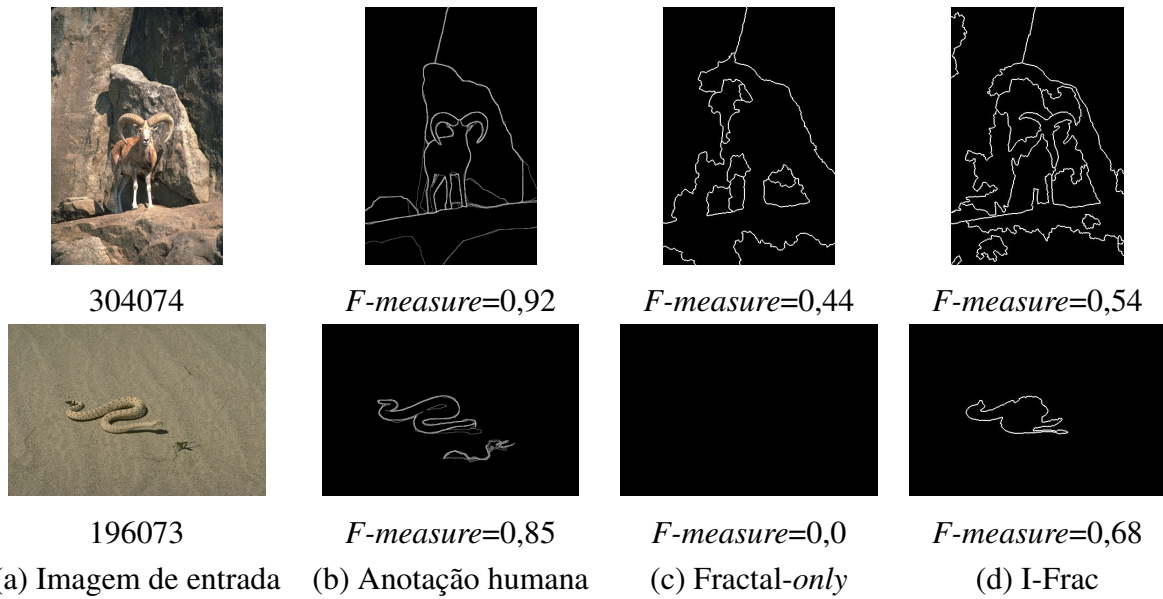


Figura 2.21: Exemplos com similaridade de cor-textura dos componentes nos quais o I-Frac obteve maiores valores de $F\text{-measure}$ que Fractal-only.

Na primeira linha da Figura 2.21, a imagem do bode nas pedras possui elemento principal e fundo de cores similares e o Fractal-only não consegue distinguir o bode. No I-Frac esse resultado melhorou, pois já é possível se distinguir a cabeça e os chifres do animal.

A segunda linha da Figura 2.21, que foi o pior resultado do Fractal-only, mostra a imagem da cobra no deserto, onde animal e ambiente são de cores similares. No Fractal-only, não houve sequer um pixel de fronteira. No I-Frac o resultado melhorou, e é melhor que o resultado do JSEG. Estes dois resultados são sobrepostos à imagem original e mostrados na Figura 2.22, para melhor visualização. O método JSEG segmenta mais a sombra do animal, e não propriamente seu corpo. Já o I-Frac consegue definir bem o contorno do animal, além de segmentar a sombra.

O I-Frac, sendo mais sensível às cores, não pode perder a generalização de texturas complexas que há no Fractal-only. Na Figura 2.23 mostra-se os resultados para imagens de entrada com estas características. Na imagem do prédio refletido na água, o resultado do

(a) $F\text{-measure}=0,54$ (b) $F\text{-measure}=0,68$

Figura 2.22: Resultados para a imagem 196073 (a) Resultado do Fractal-JSEG sobreposto à imagem original (b) Resultado do I-Frac sobreposto à imagem original

I-Frac apresentou algumas regiões a mais (algumas janelas na sombra das folhagens). Em compensação, a segmentação da folhagem na parte superior da imagem ficou mais parecida com a referência humana. No resultado da onça, o I-Frac continua identificando a textura, embora o contorno da onça apresente falhas, em compensação, o I-Frac consegue distinguir a onça do galho. Em termos semânticos, este resultado é mais parecido com a referência humana. Porém, devido às falhas no contorno, o valor de $F\text{-measure}$ diminuiu com relação ao Fractal-only.

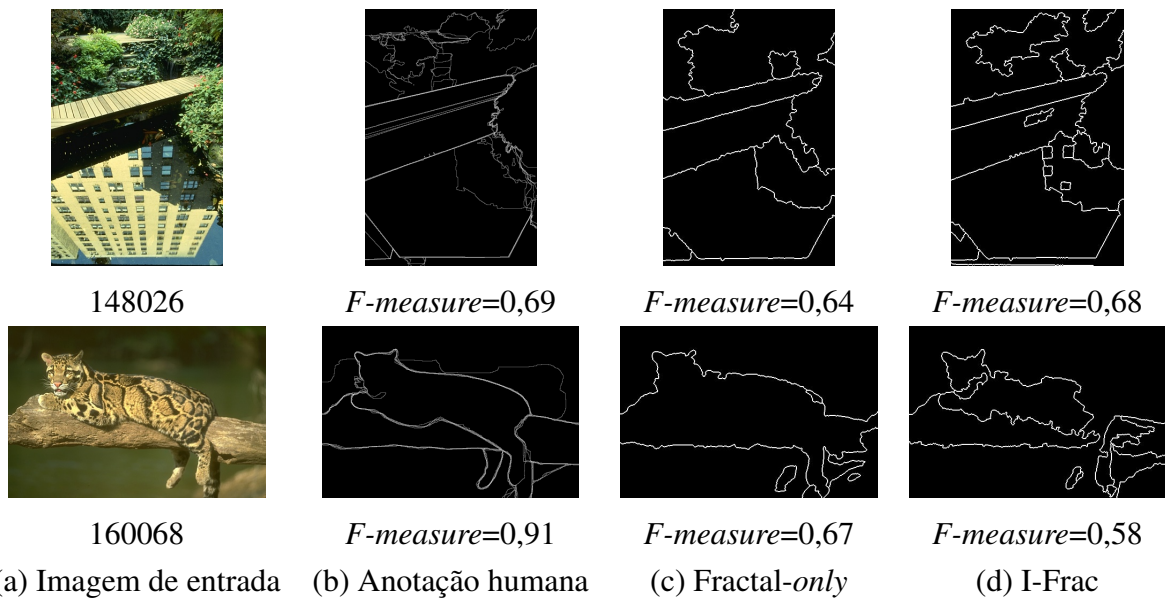


Figura 2.23: Exemplos de imagens com texturas complexas.

2.5.4 Comparação I-Frac x (JSEG, Fractal-JSEG e Fractal-only)

Os resultados do I-Frac são bem interessantes, pois manteve-se a generalização e melhorou-se a sensibilidade às cores. I-Frac também apresenta menos regiões que o JSEG e o Fractal-JSEG. Mostra-se mais um bom resultado do I-Frac a seguir, na Figura 2.24, onde o resultado final ficou bem próximo da referência humana. Houve a identificação da fronteira entre as regiões de floresta e a neve, além do contorno do lobo e, ao mesmo tempo, a floresta foi reconhecida como uma região única, não segmentando as árvores que a compõem. Além disso, os contornos são mais precisos quando comparados ao referencial humano. Porém, perdeu-se o contorno do graveto na neve, na parte esquerda da imagem, que aparecia no resultado do JSEG.

A segunda imagem da Figura 2.24 é uma cena complexa, a ponto de apresentar um *F-measure* baixo, inclusive para a referência humana. Este baixo valor significa que houve muita diferença nas segmentações manuais entre os próprios seres humanos. O elemento principal da imagem, a onça, foi o único componente em comum de todas as anotações. Assim, pode-se ver que o resultado do I-Frac é mais próximo ao referencial humano qualitativamente, embora quantitativamente, apresente um valor de *F-measure* menor que o resultado do JSEG.

Analisando os gráficos de *precision* (Figura 2.25a) e *recall* (Figura 2.25b), vê-se que o JSEG apresenta melhor *recall*, mas tem os piores resultados de *precision*. O método Fractal-only apresenta melhor resultado em termos de *precision*, mas falha em termos de *recall*. Os métodos Fractal-JSEG e I-Frac apresentam resultados entre JSEG e Fractal-only nas métricas de *recall* e *precision*. I-Frac é melhor que Fractal-JSEG porque o máximo global do I-Frac nestes dois gráficos é maior que o máximo global do Fractal-JSEG, que por coincidência ocorrem para o valor 0,7 no eixo *x*, em ambos os gráficos.

Quanto maior for o máximo valor da curva e quanto mais à direita for esse máximo, melhor é o algoritmo. No gráfico de *F-measure* (Figura 2.25c), I-Frac é melhor que todos os outros, e com uma pequena diferença do segundo melhor, o Fractal-JSEG.

Fazendo-se uma comparação par a par de métodos, com o I-Frac, e considerando-se apenas a métrica *F-measure* das imagens da base de testes, tem-se que:

- I-Frac foi melhor que o JSEG em 66% das imagens;
- I-Frac foi melhor que o método Fractal-only em 81% das imagens, e
- I-Frac foi melhor que o Fractal-JSEG em 57% dos casos.

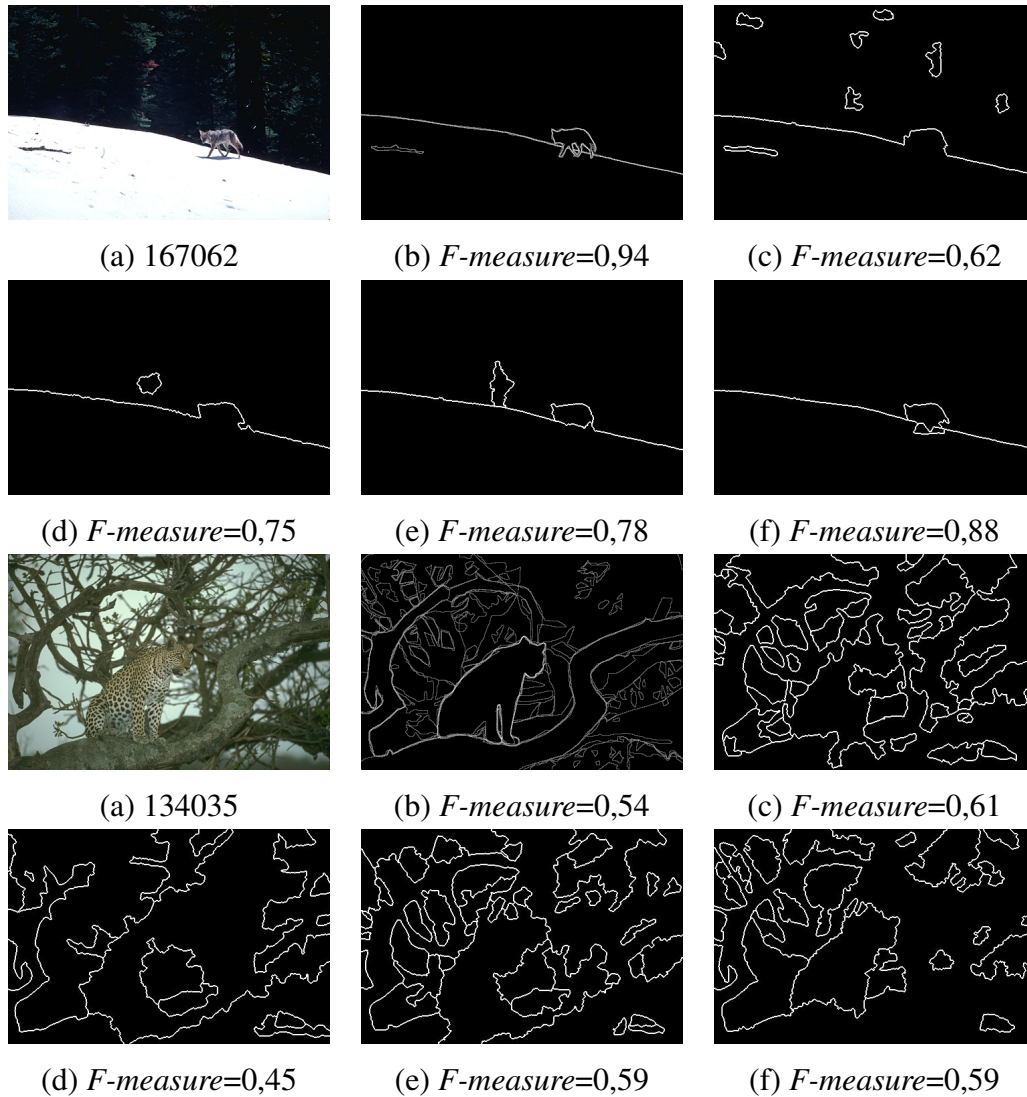


Figura 2.24: Exemplos em que o método I-Frac aumentou os valores de $F\text{-measure}$. (a) Imagem de entrada (b) Anotação humana (c) JSEG (d) Fractal-only (e) Fractal-JSEG (f) I-Frac.

2.5.5 Comparação MM-Frac x (JSEG, I-Frac e Fractal-JSEG)

A arquitetura MM-Frac utiliza duas medidas de homogeneidade de regiões: $J\text{-value}$ e um descritor multifractal. Ambas as medidas recebem a imagem original e retornam imagens que podem ser interpretadas como mapas 3D. Para integrar as duas medidas, utilizou-se uma média ponderada baseada na propriedade estatística global: o valor α . O resultado desta integração é um mapa 3D final, o qual é passado como entrada do método de crescimento de regiões.

Alguns resultados são mostrados na Figura 2.26. Foram escolhidas três imagens: uma com valor de α próximo de 3, outra com o valor 2 e outra com valor mais próximo de 1.

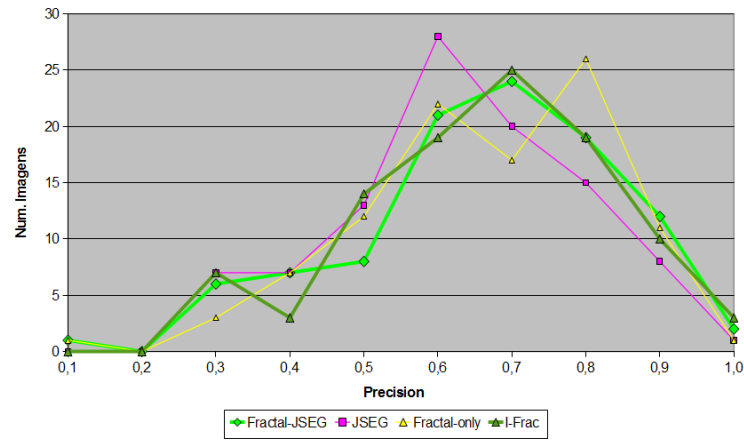
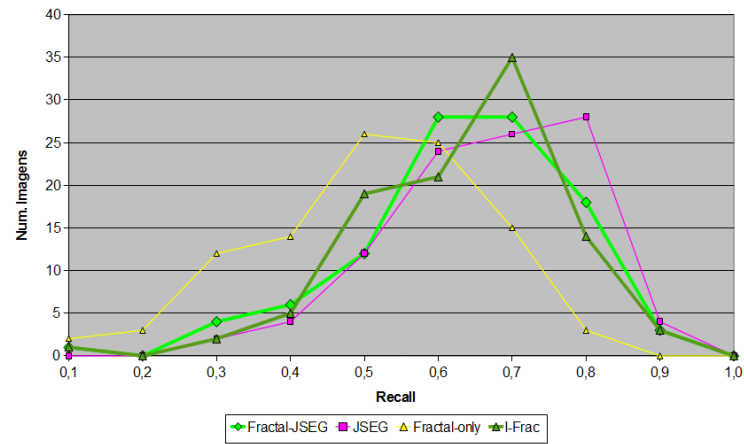
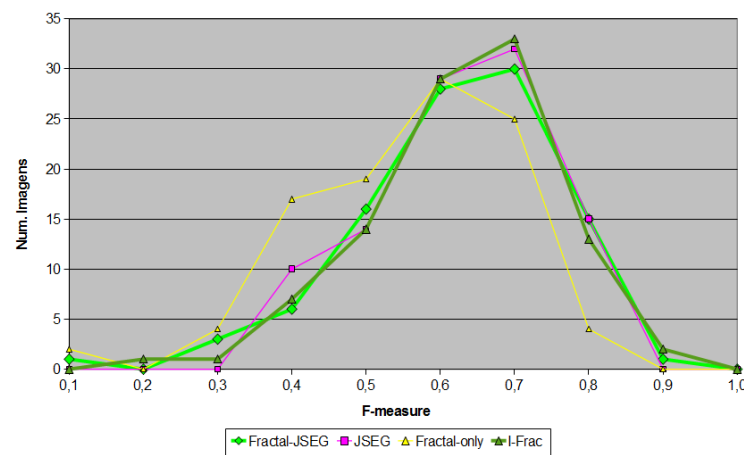
(a) Histograma da métrica *precision*.(b) Histograma da métrica *recall*.(c) Histograma da métrica *F-measure*.

Figura 2.25: Histogramas das métricas dos métodos JSEG \times Fractal-only \times Fractal-JSEG \times I-Frac.

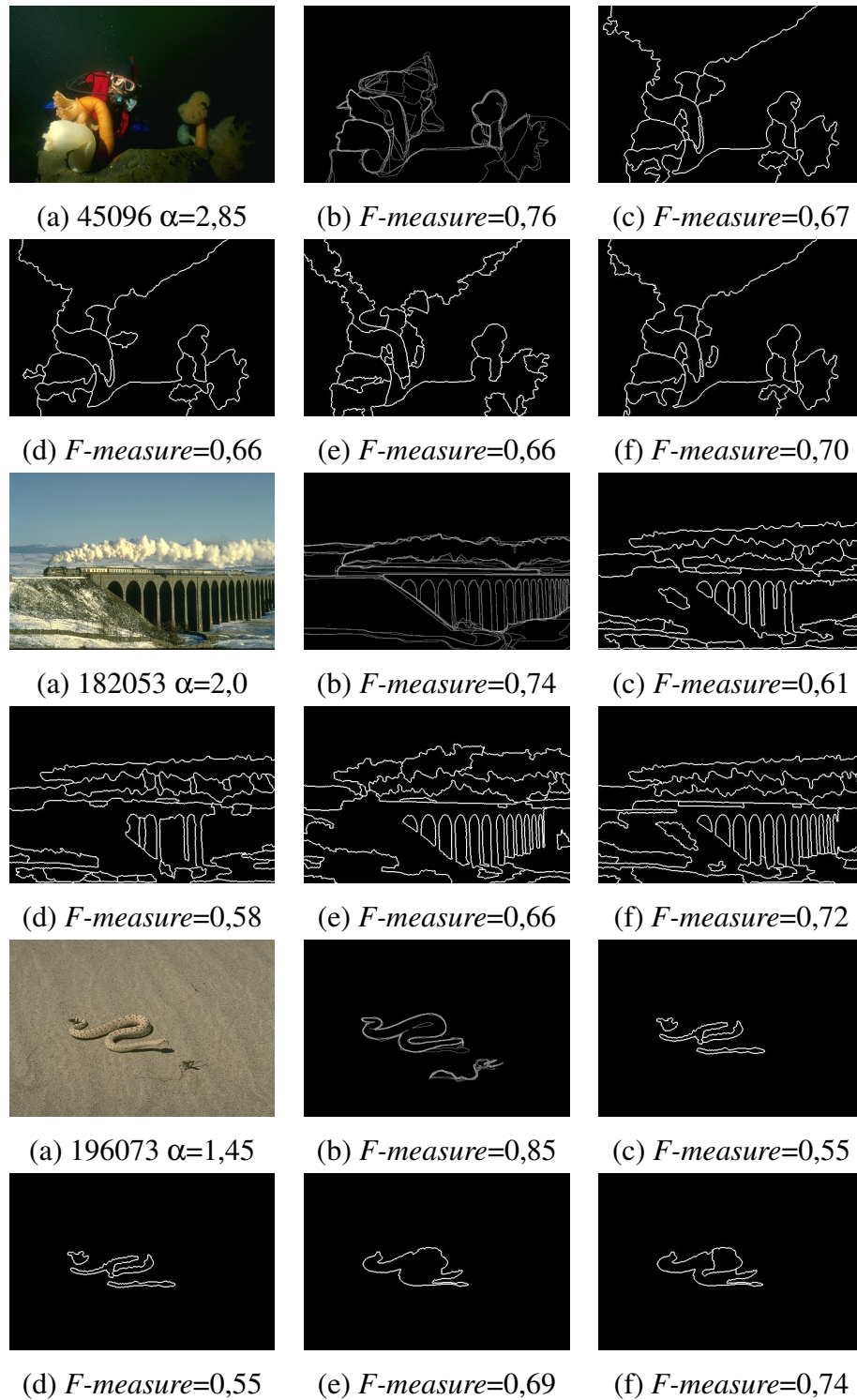


Figura 2.26: Exemplos de resultados do método MM-Frac, na ordem: (a) Imagem de entrada (b) Referência humana e as imagens resultados dos métodos (c) JSEG (d) Fractal-JSEG (e) I-Frac (f) MM-Frac.

Para estas imagens da abordagem MM-Frac, o resultado da métrica *F-measure* do método MM-Frac é maior com relação aos demais métodos. É difícil identificar qualitativamente que houve melhoria de segmentação na primeira imagem, pois os resultados visualmente são muito similares. Pelo valor alto de α , o mapa 3D final tende a ser mais parecido com o resultado do mapa do *J-image*, e com isso, é esperado que o resultado final do MM-Frac seja mais próximo do resultado do JSEG do que o do I-Frac. Para a terceira imagem, o resultado do MM-Frac é mais parecido com o resultado do I-Frac do que com o do JSEG.

No entanto, é importante frisar que o resultado não é idêntico, pois α não é igual a 3 (ou exatamente igual a 1) e, portanto, o mapa final apresenta componentes de ambas as medidas. Além disso, o método de crescimento de regiões é um método não-linear e pode ter resultados que não são simplesmente a soma linear dos resultados dos critérios independentes.

Na segunda imagem, a 182053, o resultado do MM-Frac teve melhorias com relação aos arcos da ponte (característica similar ao resultado do I-Frac) e o contorno da fumaça do trem ficou mais suave (como o resultado do JSEG), e com isso, ao final se obteve resultado de *F-measure* próximo ao do ser humano. Qualitativamente, o resultado geral do MM-Frac foi melhor que o dos outros sistemas.

Os histogramas das métricas são apresentados na Figura 2.27. O método MM-Frac acaba apresentando a curva de *F-measure* um pouco melhor que os outros métodos, pois sua curva possui o pico um pouco mais alto, além de estar um pouco mais inclinado à direita que os outros métodos.

Para mais uma comparação, os resultados das métricas dos sistemas foram tabulados na Tabela 2.3. Apesar da diferença quantitativa ser pequena, houve melhoria numérica em relação ao *F-measure* para os sistemas I-Frac e MM-Frac.

Tabela 2.3: Métricas calculadas pelo BSDS: referência humana, JSEG, Fractal-*only*, Fractal-JSEG, I-Frac e MM-Frac.

	<i>Recall</i>	<i>Precision</i>	<i>F-measure</i>
JSEG	0,61	0,56	0,58
Fractal- <i>only</i>	0,46	0,61	0,52
Fractal-JSEG	0,58	0,59	0,58
I-Frac	0,58	0,61	0,59
MM-Frac	0,63	0,56	0,59
Referência humana	0,70	0,89	0,79

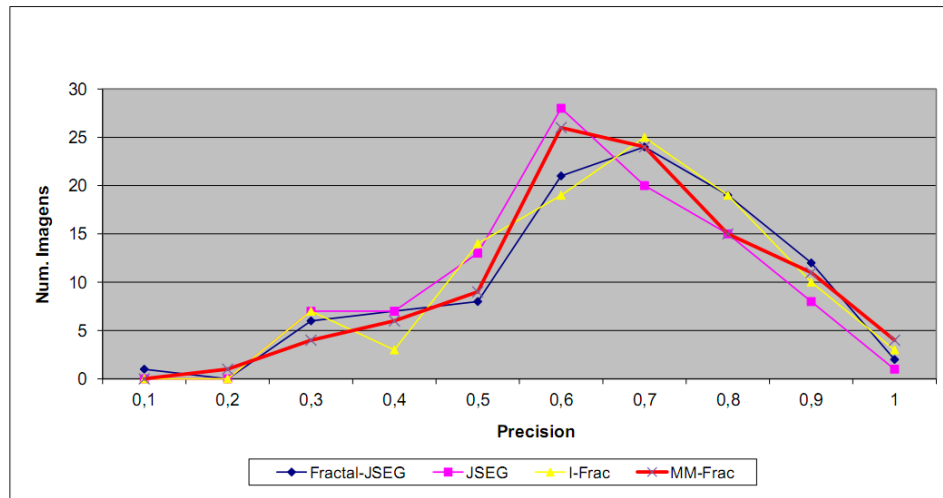
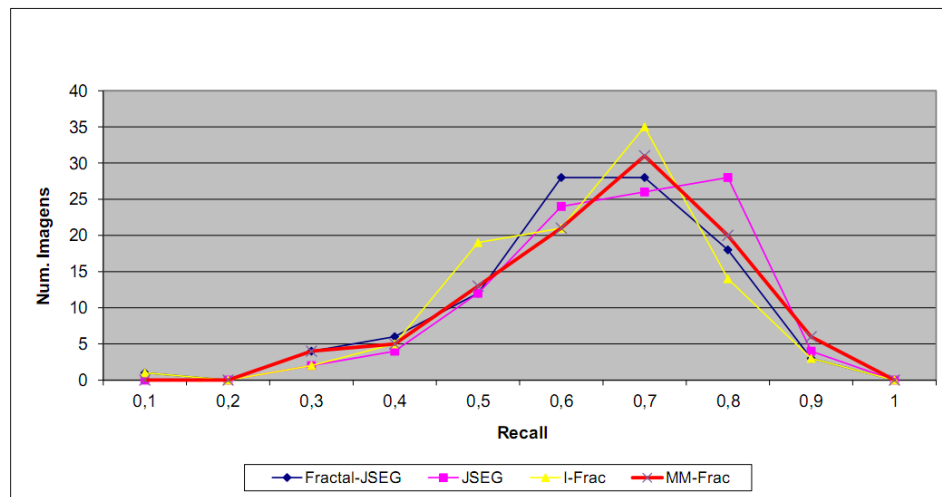
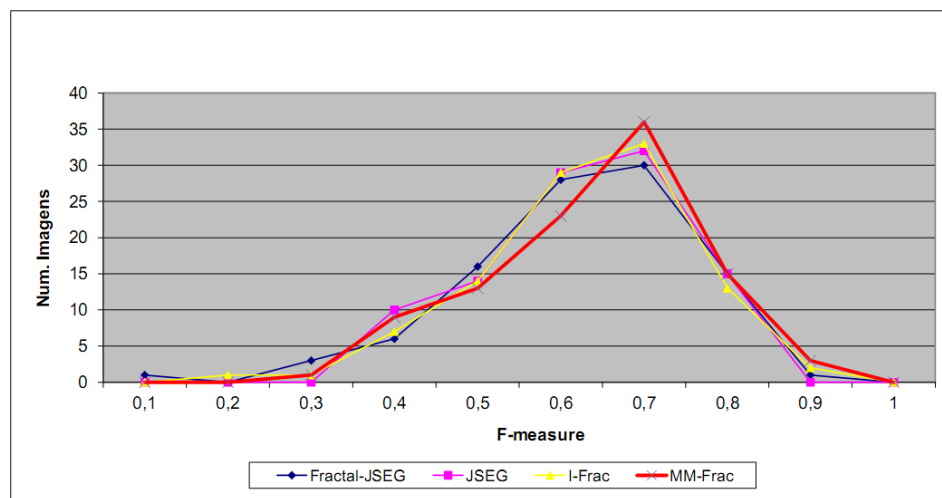
(a) Histograma da métrica *precision*.(b) Histograma da métrica *recall*.(c) Histograma da métrica *F-measure*.

Figura 2.27: Histogramas das métricas dos métodos JSEG \times Fractal-JSEG \times I-Frac \times MM-Frac.

Fazendo-se uma comparação por par de métodos, o resultado é que:

- MM-Frac foi melhor que o JSEG em 67% das imagens.
- MM-Frac foi melhor que o método I-Frac em 68% das imagens; e
- MM-Frac foi melhor que o Fractal-JSEG em 75% dos casos.

Houve apenas 7 entradas nos quais os resultados do MM-Frac tiveram valores de *F-measure* menores que ambas as entradas, e 39 casos nos quais os valores de *F-measure* foram maiores que ambas as entradas.

2.5.6 Problemas em Aberto do MM-Frac

Um problema que permanece em todos os métodos é mostrado nos resultados da imagem 42049, na Figura 2.28a: nos cantos da imagem há uma gradação suave de cores sem uma fronteira óbvia. A percepção dos anotadores humanos não fragmenta a imagem em regiões distintas (Figura 2.28b). No entanto, os resultados dos métodos automáticos vistos até o momento nesta Tese (JSEG, Fractal-only, Fractal-JSEG, I-Frac e MM-Frac) apresentam uma região quase circular na imagem (Figura 2.28c, d, e, f, g). Esta característica nas imagens já havia sido observada nos resultados do JSEG e reportado por Deng e Manjunath (2001).

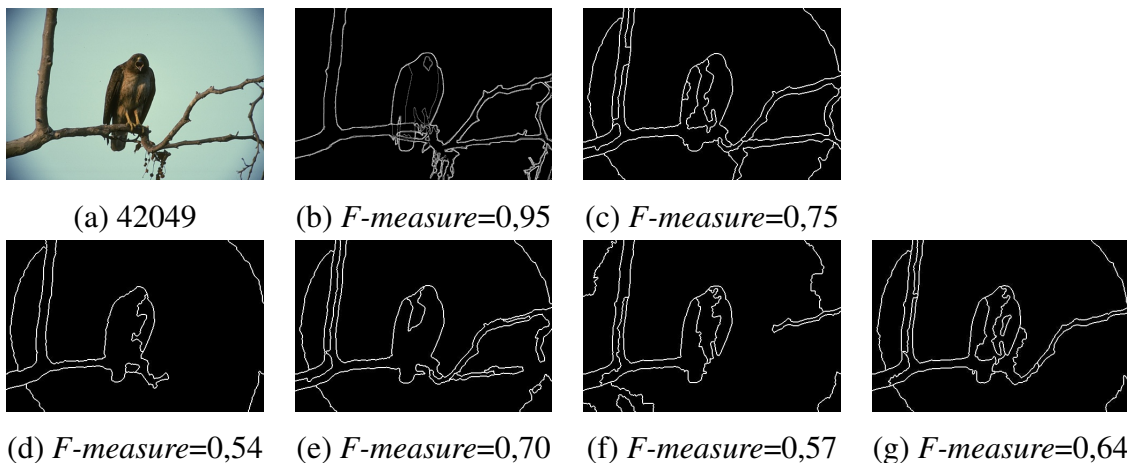


Figura 2.28: Exemplo característico do problema em aberto do MM-Frac. (a) Imagem de entrada (b) Referência humana e as imagens resultados dos métodos (c) JSEG (d) Fractal-only (e) Fractal-JSEG (f) I-Frac (g) MM-Frac.

Outro problema é a perda de componentes mais finos das imagens, como as partes do galho na parte inferior direita da imagem 42049, por exemplo. Neste exemplo, a perda é

vista nos resultados dos quatro métodos: *Fractal-only*, *Fractal-JSEG*, *I-Frac* e *MM-Frac*, sendo mais grave no resultado do *I-Frac*, que acaba por influenciar o resultado do *MM-Frac*. Nota-se que não existe este problema no método *JSEG*. Este problema é a principal causa de imagens resultantes do *MM-Frac* terem valores de *F-measure* menores que o *JSEG*. Essa característica, de generalizar texturas complexas, é uma desvantagem do descritor multifractal na imagem da águia, mas é uma vantagem na segmentação dos galhos da árvore da Figura 2.29.

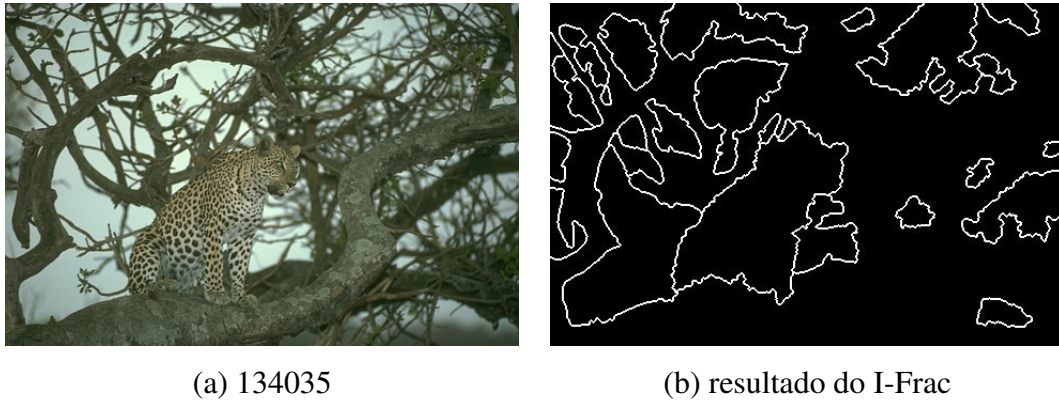


Figura 2.29: Exemplo de generalização de textura complexa do I-Frac.

2.6 Considerações Finais

Ao longo deste capítulo, foram propostas quatro estratégias, *Fractal-JSEG*, *Fractal-only*, *I-Frac* e *MM-Frac*, que embutem um descritor multifractal em um método clássico de crescimento de regiões, multiescala, chamado *JSEG*.

A motivação do *Fractal-only* foi analisar o comportamento do sistema apenas com o critério multifractal, fornecendo dados para análise de qual é a sua influência quando utilizado sozinho como critério de homogeneidade. Como se pode observar, *Fractal-only* pode fornecer resultados muito bons para algumas imagens, principalmente para segmentar texturas complexas em apenas uma única região, mas seu desempenho global é pior que o do *JSEG*.

A combinação de informações na arquitetura *Fractal-JSEG* mostrou que os resultados apresentam menos regiões que o *JSEG*, e são mais estáveis que o *Fractal-only*, no geral, melhorando o resultado quanto à referência humana.

A generalização do *Fractal-only*, de não perceber muito bem regiões de cores similares, motivou a criação de uma abordagem que fosse mais sensível a cores, o *I-Frac*. É bem verdade que toda quantização perde informação. Na imagem da cobra no deserto, por exemplo,

o resultado da quantização apresenta apenas duas cores diferentes. Logo, o uso da imagem original como entrada do método de cálculo do descritor multifractal foi de extrema importância para a melhoria da sensibilidade de cores. Com este novo critério de homogeneidade, se obteve mais sensibilidade às variações de cor, sem perder o entendimento sobre texturas complexas, e o caso relativo às imagens com cores similares entre componentes da imagem melhorou muito quando comparado à referência humana.

Uma nova mistura das duas medidas, desta vez baseada em uma informação global, o MM-Frac, mostrou que os resultados, no geral, são melhores que os métodos anteriores. Diminuiu-se a supersegmentação existente em áreas de mesma textura, tais como áreas de vegetação, distinguiu-se melhor texturas complexas, identificaram-se texturas de granularidade grossa e localizaram-se melhor as fronteiras entre regiões. No entanto, ainda persistem problemas, como a segmentação de áreas com gradação suave e a perda de contorno de informações finas.

A contribuição desta parte do trabalho foi o estudo empírico de diferentes medidas, o descritor baseado em medida multifractal e o *J-image*, como critérios de homogeneidade para um algoritmo de crescimento de regiões. Outra contribuição foi o estudo da combinação destes diferentes critérios de homogeneidade. Ao final, observou-se que o uso combinado do descritor multifractal com o *J-image* através de um controle de combinação local baseado em informação global da imagem (a arquitetura MM-Frac) melhorou os resultados da segmentação, tanto qualitativamente quanto quantitativamente, tornando-os mais próximos à percepção humana que os resultados dos métodos individuais. Esta, sem dúvida, é uma contribuição significativa desta Tese.

Capítulo 3

Integrando Crescimento de Regiões e Detecção de Bordas

Segundo a abordagem adotada por Gonzalez e Woods (2001), os métodos de segmentação são classificados em três categorias básicas: limiarização, segmentação baseada em detecção de descontinuidades e segmentação baseada em regiões. A limiarização está intimamente relacionada ao conceito de histograma. Basicamente, é estabelecido um limiar (ou limiares) que divide(m) o histograma, separando os pixels em região(ões). A dificuldade deste método é a escolha do(s) limiar(es).

A segmentação baseada em detecção de descontinuidades envolve a localização de regiões da imagem onde a variação dos tons de cinza ocorre de maneira relativamente abrupta. Enquanto os métodos de limiarização e de detecção de bordas resolvem o problema de segmentação encontrando diferenças nas tonalidades dos pixels ou conjuntos de pixels, os métodos baseados em regiões abordam a procura de similaridades entre elas. Neste caso, a segmentação é realizada unindo-se regiões adjacentes, se as mesmas forem similares, ou dividindo uma região em regiões menores, se a mesma não é homogênea (Ventura, 2009).

Os resultados de detectores de borda apresentam lacunas e informações ruidosas, enquanto os resultados de métodos de crescimento de regiões tendem a apresentar supersegmentação, além de gerar contornos imprecisos e falsos (Muñoz et al., 2003). A estratégia do método de integração proposto neste capítulo é tirar proveito da natureza complementar destes dois métodos, isto é, retirar os contornos falsos das regiões através da informação de bordas do resultado do detector de borda, e retirar o ruído da informação de bordas guiado pelos contornos do resultado do algoritmo de crescimento de regiões, integrando as duas entradas em um único resultado como saída.

Inicia-se o capítulo com uma subseção listando trabalhos correlatos, seguida de outra

subseção explicando como foi a escolha do método de detecção de bordas. Em seguida, são descritas duas propostas de estratégias automáticas e não-supervisionadas de integração dos resultados da arquitetura MM-Frac (apresentado no capítulo anterior) com os resultados de um método clássico de detecção de bordas. As estratégias se chamam: KSS (referente aos nomes **K**omati, **S**alles e **S**arcinelli) (Komati et al., 2011b,c,a), e KoSS (agora **K**omati, **S**alles e **S**arcinelli) (Komati et al., 2011d). São apresentados vários experimentos, e os resultados são discutidos ao final do capítulo.

3.1 Trabalhos Relacionados

Existem várias propostas combinando os resultados de detecção de bordas e métodos orientados a regiões. Munõz e colegas (Muñoz et al., 2003) fazem uma revisão de vários métodos e os classificam em duas classes: *embedded integration* e *post-processing integration*. *Embedded integration* produz, em geral, um único algoritmo complexo que evita erros no resultado final. A estratégia *post-processing integration* trabalha com um conjunto de algoritmos, e aceita falhas nos resultados dos mesmos. É o módulo de integração *a posteriori* que será responsável por corrigir estes erros. A estratégia proposta neste capítulo é classificada, segundo esta ótica como *post-processing integration*. As informações de regiões e de bordas são extraídas separadamente e depois integradas. Munõz e colegas (Muñoz et al., 2003) identificaram três diferentes objetivos de integração:

1. refinamento de contornos: o resultado da segmentação por regiões é considerado como ponto de partida, e a informação do resultado do método de detecção de bordas é usado para alterar a forma dos contornos das regiões;
2. seleção do melhor: a informação de borda é usada para avaliar a qualidade de vários resultados de segmentação de regiões; e
3. *over-segmentation*: esta abordagem consiste em eliminar contornos falsos.

No refinamento de contornos há duas técnicas: multi-resolução e *snakes*. Na técnica de multi-resolução, o resultado da segmentação por regiões é considerado como uma segmentação grosseira (*coarse*), e é refinado aumentando a resolução em diferentes escalas, usando uma estrutura piramidal ou *quadtree*. Para cada nova escala, refina-se a localização dos contornos, guiados pela informação do resultado da detecção de bordas. Depois, faz-se o processo inverso, diminuindo a resolução, e gera-se a imagem segmentada final. Destacam-se os trabalhos de Spann e Wilson (1985, 1988, 1990) e Hsu e colegas (Hsu et al., 2000).

O método *snakes*, ou *active contour*, é um modelo para delinear o contorno externo de objetos numa imagem. *Snakes* é um método de minimização de energia, onde cada contorno é um *spline* deformável, influenciado por forças que puxam/empurram, como esticar/puxar elásticos. O resultado do crescimento de regiões é considerado como ponto de partida e o resultado da detecção de bordas é utilizado para o cálculo de energia. Listam-se alguns trabalhos nesta linha: Saeed e Dugelay (2010), Sclaroff e Liu (2001) e Chan et al. (1996).

Um método não abordado pelo trabalho de Munõz e colegas (Muñoz et al., 2003), é o refinamento de contornos usando o método *multi-label graph-cut* (como mostrado em Komati et al. 2011d).

A seleção do melhor ocorre quando o algoritmo de segmentação por regiões gera diferentes resultados através da mudança de parâmetros. Uma função de avaliação mede a qualidade destes vários resultados, de acordo com a consistência do resultado da detecção de bordas. De forma simplificada, a melhor segmentação é aquela na qual os contornos das regiões são mais próximas das bordas resultantes do algoritmo de detecção de bordas. Um trabalho pioneiro foi o de Fua e Hanson (1987), e um trabalho mais recente é o de Revol-Muller e colegas (Revol-Muller et al., 2000).

A abordagem *over-segmentation* (Fjørtoft et al., 1997; Gagalowicz e Monga, 1986) surgiu como resultado da dificuldade em estabelecer um critério de homogeneidade “perfeito” para o algoritmo de crescimento de regiões. Como Pavlidis e Liow (1990) sugeriram, a principal razão do algoritmo de crescimento de regiões produzir contornos falsos é que a definição de uniformidade de uma região é muito rigorosa. Um exemplo é a iluminação de uma imagem, onde o brilho pode variar linearmente dentro de uma região, e, em geral, critérios de homogeneidade enxergam estas variações como não similares. Assim, é muito difícil encontrar critérios de uniformidade que correspondam exatamente a essas condições e não gerem fronteiras falsas. Eles concluíram que os resultados poderiam ser significativamente melhorados, verificando se as fronteiras de todas as regiões também são bordas no resultado do método de detecção de bordas, ao invés de refinar os critérios de homogeneidade.

No trabalho de Pavlidis e Liow (1990), é utilizado o método *split-and-merge*, e para cada fronteira entre regiões deste resultado calculou-se uma “função de mérito”. A função de mérito é uma soma de duas parcelas: uma é a soma do contraste dos pixels (que estão na mesma posição dos pixels de fronteira do resultado do *split-and-merge*) do resultado da detecção de bordas, e outra é uma soma de mudanças de direções ao longo da fronteira. De acordo com o trabalho, o resultado do *split-and-merge* apresentado tem um aspecto bem “quadriculado” (*blocking*), e, em geral, as fronteiras falsas eram linhas retas. Utiliza-se um algoritmo exaustivo, em que cada iteração elimina a fronteira de menor valor da função de

mérito, até que se chegue a um parâmetro definido pelo programador. Assim, quanto menor a função de mérito, (isto é, não há muitos pixels de borda no resultado do detector de bordas e as fronteiras são linhas retas), maior a probabilidade desta fronteira ser eliminada. Além da eliminação das bordas falsas, também é feita uma modificação da forma do contorno das regiões.

Neste contexto, parte da proposta desta segunda integração se baseou em parte do trabalho de Pavlidis e Liow (1990), a parte a qual o objetivo é o de retirar contornos falsos. Porém, adotou-se outra função de mérito, utilizando informação global e local.

3.2 Arquiteturas Implementadas

Nesta proposta, os resultados dos métodos de detecção de bordas e crescimento de regiões são extraídos independentemente, a partir da imagem de entrada, como mostrado na Figura 3.1. Mais ainda, as estratégias a serem apresentadas são independentes de qual foi o método baseado em região ou o detector de borda. No entanto, é necessário que a imagem fornecida pelo método orientado a região seja uma imagem binária, e que a imagem fornecida pelo método de detecção de borda seja uma imagem em escala de cinza. Esta imagem em escala de cinza pode ser interpretada da seguinte forma: quanto maior o valor do pixel na escala, maior a confiança de que aquele pixel é um pixel de borda.

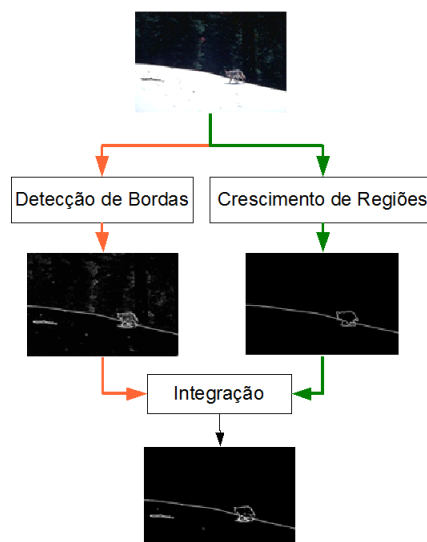


Figura 3.1: Arquitetura geral da estratégia de integração.

3.2.1 A Escolha do Detector de Bordas

O processo de detecção de bordas consiste na verificação da variação dos valores de luminosidade de uma imagem, visando localizar e realçar os pixels de borda através da ampliação do contraste entre uma região e outra região vizinha.

Foram escolhidos como potenciais candidatos para detecção de bordas alguns métodos tradicionais (Gonzalez e Woods, 2001), que resultam em imagens em escala de cinza: Sobel, Laplace, Prewitt, LoG e gradiente morfológico. Reitera-se que não foram usados métodos como o filtro de Canny, pois o resultado é uma imagem binária. Em todos os filtros espaciais utilizou-se o tamanho 3×3 . O parâmetro *alpha* do filtro de Laplace foi configurado para 0,2.

O gradiente morfológico é definido pela diferença aritmética entre a dilatação e a erosão de uma imagem por um elemento estruturante. Como a dilatação e a erosão representam, respectivamente, o máximo e o mínimo, então cada ponto do gradiente morfológico é a diferença entre os valores máximo e mínimo do ponto central do elemento estruturante. Como toda operação morfológica, o gradiente depende da forma e tamanho do elemento estruturante. Foi escolhido um elemento estruturante de tamanho 3×3 em formato de cruz (ou 4-conectado) com valores 1.

No experimento simplesmente foram aplicados os detectores de borda diretamente nas imagens em escala de cinza da base de teste do BSDS e calcularam-se as métricas *recall*, *precision* e *F-measure*. Conforme processo de uso do BSDS, descrito na Introdução (Figura 1.10).

Todos os resultados dos cinco métodos são imagens em escala de cinza, gerando uma imagem “*soft edge map*”, que é um tipo de imagem aceito pelo BSDS como imagem de segmentação. É o algoritmo do BSDS que efetua a binarização da imagem antes do cálculo das métricas. Se o parâmetro de binarização passado for 30, significa que o BSDS irá usar 30 diferentes valores de limiares para a binarização, gerando 30 diferentes imagens de segmentação. Indica-se, como resultado final, o limiar que gerou o maior valor de *F-measure*. As métricas calculadas pelo BSDS são apresentadas na Tabela 3.1. Nota-se que os valores são muito parecidos, e é difícil tomar uma decisão baseada apenas nesta tabela.

Outro resultado comparativo é apresentado na Tabela 3.2. Cada célula numérica da tabela representa a quantidade de imagens da base de testes do BSDS que teve o maior valor da métrica da coluna com relação ao método da linha. Assim, o gradiente morfológico apresentou maior valor de *recall* em 34 das 100 imagens e maior valor de *precision* em 63 das 100 imagens. Observa-se que, no que tange a *F-measure*, o gradiente morfológico obteve resultados muito acima dos demais.

Tabela 3.1: Tabela comparativa das métricas calculadas pelo BSDS nos métodos de detecção de bordas testados.

	<i>Recall</i>	<i>Precision</i>	<i>F-measure</i>
Sobel	0,58	0,35	0,44
Laplace	0,64	0,32	0,43
Prewitt	0,58	0,35	0,44
LoG	0,63	0,32	0,42
Gradiente Morfológico	0,59	0,36	0,45

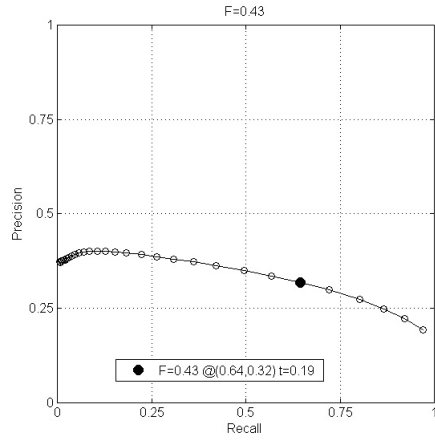
Tabela 3.2: Tabela comparativa entre os métodos de detecção de bordas, mostrando a quantidade de imagens em que cada método (linha) obteve o maior valor da métrica (coluna).

	<i>Recall</i> #imagens	<i>Precision</i> #imagens	<i>F-measure</i> #imagens
Sobel	13	11	7
Laplace	22	6	1
Prewitt	5	16	9
LoG	23	4	9
Gradiente Morfológico	34	63	74

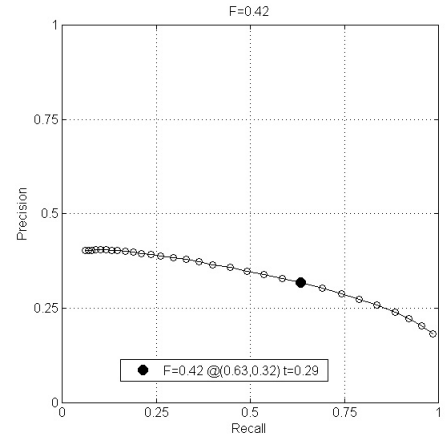
Uma comparação gráfica pode ser vista na Figura 3.2. Os algoritmos de Laplace (Figura 3.2a) e LoG (Figura 3.2b) possuem comportamento bem similares, assim como os métodos de Prewitt (Figura 3.2c) e Sobel (Figura 3.2d). Na Figura 3.2e apresenta-se o resultado do gradiente morfológico.

A comparação entre três métodos (Sobel \times Laplace \times Gradiente Morfológico) é mostrada na Figura 3.2f, onde é possível verificar que a curva do gradiente morfológico está sempre acima das outras, indicando que ele possui métricas melhores. O melhor resultado do gradiente morfológico é devido às suas características de remoção de pequenos artefatos escuros e claros e de depender menos das direções das bordas quando se usa um elemento estruturante simétrico (Gonzalez e Woods, 2001). Assim, optou-se pelo gradiente morfológico como método de detecção de bordas.

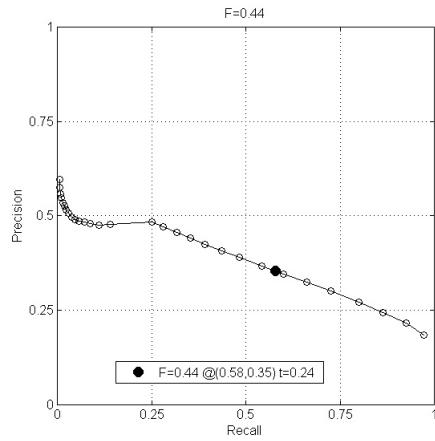
É comum usar um método de suavização como pré-processamento do detector de bordas, e com isso retirar parte do ruído. Aqui foi escolhido um método clássico não-linear do tipo “*edge-preserving smoothing*” (EPS), o filtro de Kuwahara (Kuwahara et al., 1976), com máscara de tamanho 5×5 . Para tratar imagens coloridas foram processados separadamente os canais de cores do espaço RGB, e os resultados dos três canais foram somados. Para mos-



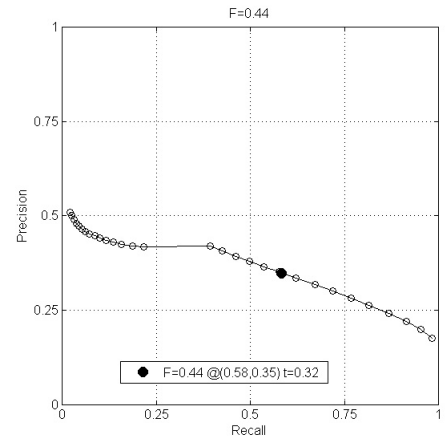
(a) Laplace



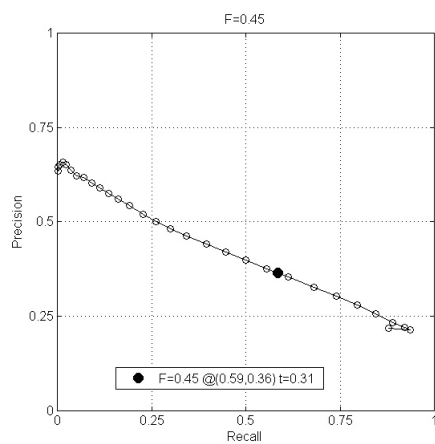
(b) LoG



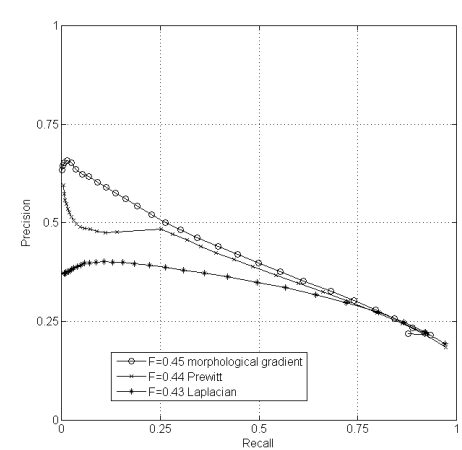
(c) Prewitt



(d) Sobel



(e) Gradiente Morfológico

(f) Prewitt \times Laplace
 \times Gradiente MorfológicoFigura 3.2: Gráficos de *Precision* \times *Recall* dos detectores de borda.

trar que houve melhoria com estes processamentos extra, o gráfico comparativo de métricas é apresentado na Figura 3.3.

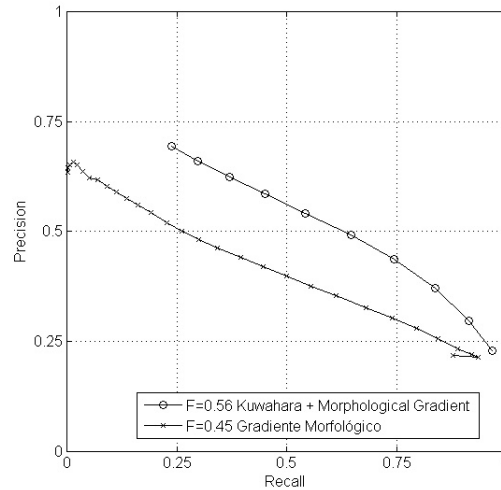


Figura 3.3: Gráficos de *Precision* \times *Recall* do Gradiente Morfológico aplicado em imagem em escala de cinza e aplicado em imagem colorida com pré-processamento via filtro de Kuwahara.

3.2.2 KSS

Para a integração dos resultados de bordas e regiões, o primeiro algoritmo a ser apresentado é o KSS (Komati et al., 2011b,c). A arquitetura geral do método está na Figura 3.4, e o pseudocódigo no Algoritmo 1. No passo 1 do algoritmo são definidos os nomes dos parâmetros de entrada. No passo 2, a soma irá enfatizar todos os pixels de fronteira que ocupam as mesmas posições nos dois mapas de entrada. A lógica do resto do algoritmo é eliminar ou reduzir as informações falsas.

No passo 3 é construído um mapa de bordas fracas baseado na informação do mapa de bordas. Note-se que o controle do KSS é baseado no resultado do detector de bordas. Quanto menos informação tiver este resultado, maior é o seu grau de confiança no método de detecção e menor é a confiança no resultado do método baseado em regiões. Neste ponto, usou-se a conjectura de que quando há muita informação numa imagem o ser humano tende a usar apenas as informações mais “grosseiras” de sua percepção, não se atendo aos detalhes. Da mesma forma, quando o detector de bordas tem muita resposta (apresenta muitos pixels com valores próximos da cor branca), confia-se mais na resposta menos detalhada do crescimento de regiões.

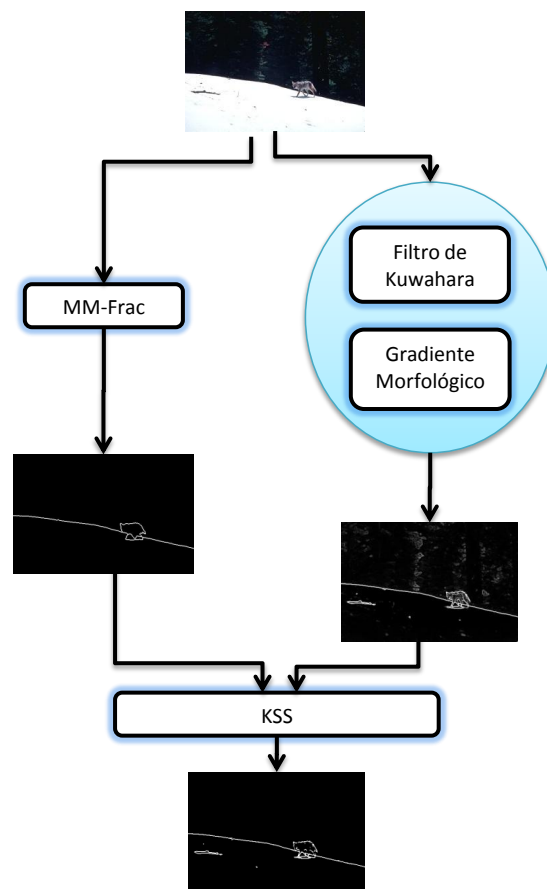


Figura 3.4: Arquitetura do KSS.

Algorithm 1 KSS

-
- 1: Imagens de entrada: mapa-de-bordas e mapa-de-regiões
 - 2: Faça a imagem-resultado = mapa-de-bordas + mapa-de-regiões
 - 3: Construa um mapa-de-bordas-fracas a partir do mapa-de-bordas
 - 4: **for** cada pixel da imagem-resultado **do**
 - 5: **if** ((se a mesma posição do pixel no mapa-de-regiões é um pixel de fronteira) E (a maior parte da vizinhança da posição do pixel está no mapa-de-bordas-fracas) **then**
 - 6: apague este pixel da imagem-resultado
 - 7: **end if**
 - 8: **if** (se a mesma posição do pixel é marcada como não-fronteira no mapa-de-regiões) E (a mesma posição do pixel está no mapa-de-bordas-fracas) **then**
 - 9: apague este pixel da imagem-resultado
 - 10: **end if**
 - 11: **end for**
-

Para automatizar o cálculo da quantidade de informação, utiliza-se o formato do histograma da imagem resultante do detector de bordas. Assim, estima-se a quantidade de pixels que indicam que não há borda (maior quantidade de pixels escuros, ou seja, valores próximos de zero - entre 0 e 50), relativo à informação de certeza média de existência de borda (valores existentes no meio do histograma, entre 50 e 200 na escala de cinza), idéia similar àquela encontrada em (Rotem et al., 2007). O valor do limiar é dado por

$$\text{limiar}_{\text{bordafraca}} = \frac{\sum_{i=0}^{50} h_i}{\sum_{i=50}^{200} h_i}, \quad (3.1)$$

onde $i = [0, 255]$ é o valor do pixel na imagem em escala de cinza, e h_i é a frequência (em y) de i (em x).

Todos os pixels de valores menores que o $\text{limiar}_{\text{bordafraca}}$ são considerados de borda fraca. Quanto menos informação tiver esta imagem, maior é o valor de limiar e se tem mais pixels de borda fraca. Assim, este passo é basicamente a criação de uma outra imagem através de um processo de binarização a partir da informação global do resultado do detector de bordas.

A Figura 3.5 mostra dois exemplos de valores de limiares opostos. Na primeira linha tem-se a imagem de entrada, na segunda linha o resultado do detector de bordas, na terceira linha o histograma da imagem resultante do detector de bordas e, por último, é mostrado o mapa-de-bordas-fracas. A coluna esquerda mostra um exemplo em que o limiar é bem alto e, logo, existem muitos pixels que são considerados como bordas fracas. Bordas fracas são pixels brancos, e logo quase toda a imagem fica em branco. A coluna direita mostra um exemplo oposto, com um valor bem mais baixo de limiar e, assim, quase não existem pixels considerados de borda fraca.

Nos passos 4-11, trabalha-se com duas imagens binárias: o mapa-de-bordas-fracas (resultado do passo 3) e o mapa-de-regiões (fornecido pelo MM-Frac). O comando de condição (passos 5, 6 e 7) elimina fronteiras falsas provenientes do mapa de regiões. O comando de condição das linhas 8, 9 e 10 elimina a informação inerente da rugosidade da textura (que não são fronteiras entre regiões) proveniente da imagem de entrada, o mapa-de-bordas. Quando maior o limiar, mais se confia no resultado do detector de bordas e, com isso, eliminam-se mais informações provenientes das imagens de entrada.

Quanto menor o $\text{limiar}_{\text{bordafraca}}$, como no caso do coala (a coluna direita da Figura 3.6), os passos 4-11 do algoritmo praticamente não são executados. Neste caso, o resultado melhora devido ao comando do passo 2, que enfatiza todos os pixels de borda que estão na mesma posição em ambas as imagens de entrada e ao mesmo tempo diminui a relevância dos



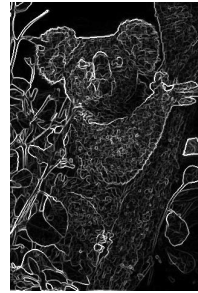
(a) 42049



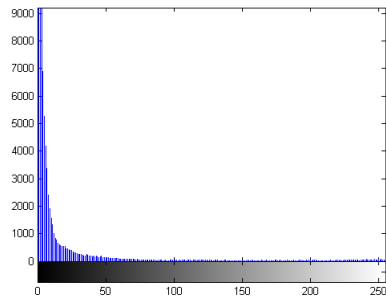
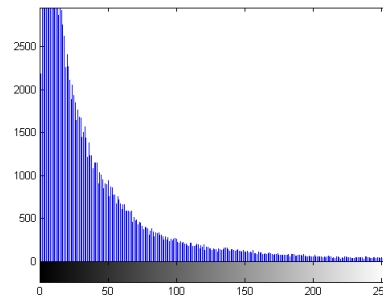
(b) 69015



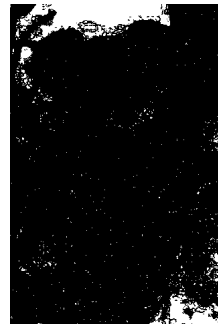
(c)



(d)

(e) $\text{limiar}_{\text{bordafraca}}=24,2$ (f) $\text{limiar}_{\text{bordafraca}}=2,9$ 

(g)



(h)

Figura 3.5: Exemplos de imagens para um baixo e alto valor de $\text{limiar}_{\text{bordafraca}}$. (a) e (b) imagens de entrada; (c) e (d) imagens provenientes do detector de bordas; (e) e (f) histogramas das imagens (c) e (d), respectivamente, e (g) e (h) imagens referentes ao mapa-de-bordas-fracas.

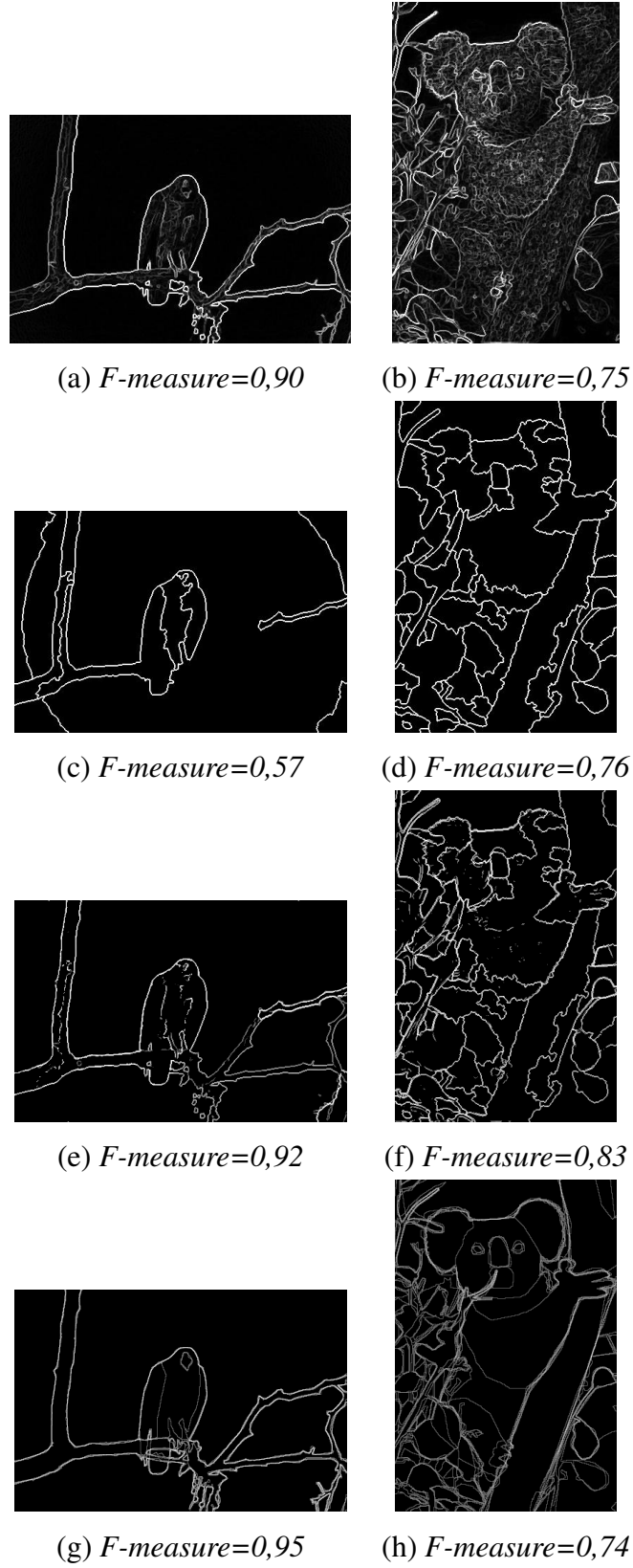


Figura 3.6: Imagens de entrada e saída do KSS. Imagens (a) e (b) resultantes do detector de bordas; (c) e (d) resultantes do MM-Frac; (e) e (f) resultantes do KSS e, (g) e (h) anotações humanas.

pixels fracos provenientes do mapa-de-bordas. Embora qualitativamente seja difícil verificar a melhoria do resultado do KSS (Figura 3.6f) com relação ao método MM-Frac (Figura 3.6d), é fácil verificar que o valor de *F-measure* melhorou. Também é fácil notar que o resultado do KSS não manteve o ruído do resultado da detecção de bordas (Figura 3.6b).

Na coluna esquerda da Figura 3.6), temos o exemplo no qual o $\text{limiar}_{\text{bordafraca}}$ é bem alto. Neste caso, devido aos passos 5, 6 e 7, o contorno circular nos cantos do resultado do MM-Frac (Figura 3.6c) serão apagados do resultado do KSS (Figura 3.6e), pois se encontram na área considerada borda fraca pelo mapa-de-bordas-fracas (Figura 3.5g). Os pixels de ruído dentro do contorno da águia do resultado da detecção de bordas (Figura 3.6a) são eliminados no resultado do KSS, devido aos passos 8, 9 e 10. Devido ao passo 2, detalhes perdidos no resultado de crescimento de regiões são resgatados pelo método de detecção de bordas, como os galhos da parte inferior direita.

É importante notar que, em ambos os casos, as imagens resultantes do KSS apresentaram valor de *F-measure* maior que as imagens de entrada, e assim, são mais próximas da anotação humana.

3.2.3 KoSS

KoSS (Komati et al., 2011d) é uma extensão do algoritmo KSS. No geral, a concepção do algoritmo não mudou: as entradas são as mesmas e a lógica continua sendo eliminar ou reduzir as informações de borda falsa provenientes do processo baseado em regiões, através da quantificação da confiança do resultado do detector de bordas e, ao mesmo tempo, enfatizar e preservar as bordas fortes que coincidem em ambas as entradas.

A diferença principal é que o KSS faz o processamento de eliminação de bordas pixel-a-pixel, enquanto o KoSS analisa cada segmento da fronteira do mapa de regiões. A melhoria do KoSS sobre o KSS é a eliminação de alguns artefatos gerados pela entrada do mapa de regiões. A Figura 3.7 apresenta resultados da segmentação de uma imagem do BSDS, em que fica evidente a diferença do processamento. A figura mostra (a) a imagem de entrada, (b) a referência humana, (c) o resultado do MM-Frac, (f) o resultado do KSS e (i) o resultado do KoSS. A imagem 3.7d é a sobreposição da imagem (c) sobre a (a), assim como a (g) é a sobreposição da imagem (f) sobre a (a), e a (j) é a sobreposição da imagem (i) sobre a (a).

A linha no céu sobre a fumaça do trem no resultado do MM-Frac (Figura 3.7e) é uma borda falsa, e é totalmente eliminada no resultado do KoSS (Figura 3.7k). Como a eliminação do KSS é feita pixel-a-pixel, a imagem resultante ainda apresenta alguns pixels perdidos no céu, mais fáceis de visualizar na imagem ampliada (Figura 3.7h). A eliminação total da borda

falsa torna o resultado mais próximo da referência humana. Os números entre parênteses são os valores de *F-measure* associados à imagem.

Na sequência, o algoritmo KoSS é apresentado como pseudo-código (ver o Algoritmo 2). As linhas 1, 2 e 3 do algoritmo KoSS têm a mesma função das linhas 1, 2 e 3 do algoritmo KSS. A diferença se encontra entre as linhas 4 à 15, Parte I do pseudocódigo. Na linha 4, a partir do mapa de regiões contrói-se uma lista de segmentos de fronteira. O mapa de região é visto como um esqueleto, cujos pontos podem ser classificados como pontos finais, normais e ramificações (Attali et al., 1997). Numa vizinhança 3×3 , os pontos finais são definidos como aqueles que só possuem um único vizinho, os pontos normais têm exatamente dois vizinhos, e pontos de ramificações têm mais que dois vizinhos. Um segmento-de-fronteira é um subconjunto de pixels que consiste totalmente de pontos normais, exceto pelos extremos, que são pontos finais ou pontos de ramificação.

Algorithm 2 KoSS

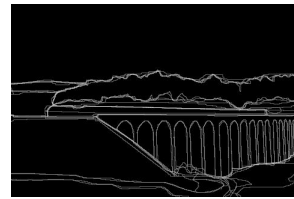
```

1: Imagens de entrada: mapa-de-bordas e mapa-de-regiões
2: Faça a imagem-resultado = mapa-de-bordas + mapa-de-regiões
3: Construa um mapa-de-bordas-fracas a partir do mapa-de-bordas
4: Construa uma lista-de-segmentos-de-fronteiras a partir do mapa-de-regiões
   {Parte I}
5: for cada segmento-de-fronteira da lista-de-segmentos-de-fronteiras do
6:   contador-de-bordas-fracas = 0
7:   for cada pixel do segmento-de-fronteira do
8:     if (a maioria da vizinhança deste pixel está no mapa-de-bordas-fracas com cor
        branca) then
9:       incremente contador-de-bordas-fracas
10:    end if
11:  end for
12:  if (contador-de-bordas-fracas > um terço do comprimento total do segmento-de-
    fronteira) then
13:    apague todo o segmento-de-fronteira da imagem-resultado
14:  end if
15: end for
   {Parte II}
16: for cada pixel marcado como não-borda no mapa-de-regiões do
17:   if (a mesma posição do pixel está no mapa-de-bordas-fracas) then
18:     apague este pixel da imagem-resultado
19:   end if
20: end for

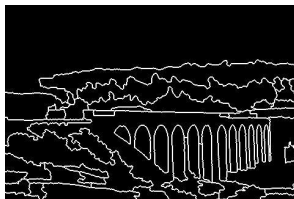
```



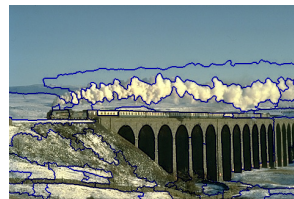
(a) Imagem 182053.



(b) Anotação humana (0,74).



(c) Resultado do MM-Frac (0,69).



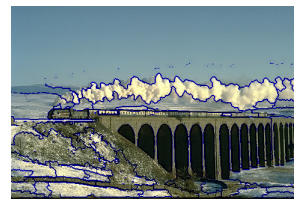
(d) MM-Frac + imagem de entrada.



(e) Ampliação da borda falsa de (c).



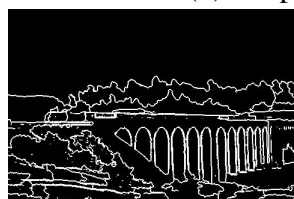
(f) Resultado do KSS (0,70).



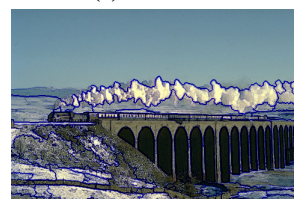
(g) KSS + imagem de entrada.



(h) Ampliação do artefato de (f).



(i) Resultado do KoSS (0,71).



(j) KoSS + imagem de entrada.



(k) Ampliação da eliminação da borda falsa de (i).

Figura 3.7: $KSS \times KoSS$: artefatos no resultado do KSS, mas não no do KoSS.

O Algoritmo 3 mostra o código em Matlab correspondente à criação desta lista de segmentos de fronteiras. Neste código, o objetivo da linha 1 é encontrar todos os pontos de ramificação, enquanto que a linha 2 apaga estes pontos da imagem. Com isso, tem-se uma imagem com vários segmentos de fronteira, isto é, tem-se vários pontos normais conectados entre pontos finais. Na linha 3, a função **regionprops** faz todo o trabalho e retorna uma lista dos segmentos de fronteira existentes na imagem, na variável **lineStats**.

Algorithm 3 Linha 4 do Algorithm KoSS em Matlab

```
1: Ipoints = bwmorph(MMFracImage, 'branchpoints', 1);
2: MMFracImage(Ipoints) = 0;
3: lineStats = regionprops(MMFracImage, 'PixelList', 'PixelIdxList');
```

Continuando no Algoritmo 2, verifica-se a quantidade de bordas fracas presentes em cada segmento. Se a contagem for maior que pelo menos um terço do tamanho do segmento, todo o segmento será apagado do resultado final (o valor de um terço foi alcançado através de testes).

A Parte II (linhas 16 a 20) do pseudocódigo do Algoritmo 2 é exatamente o comando de condição das linhas 4 e 8 a 11 do Algoritmo 1. Isto é, tenta-se eliminar parte da informação proveniente do mapa-de-bordas, referente à rugosidade da textura.

3.3 Resultados Experimentais e Discussão

No ambiente de testes adotado é importante frisar que não houve fase de treinamento ou ajuste de parâmetros, e todos os métodos foram executados de forma automática, sem intervenção humana ou parametrização específica para qualquer das imagens. Os parâmetros do MM-Frac são os mesmos do método JSEG, e portanto não necessitam de definição de valores pelo usuário. O gradiente morfológico segue os parâmetros com os quais foi testado, isto é, utiliza vizinhança de 3×3 , e o método de Kuwahara usou janela de tamanho 5×5 . O método KSS não possui parâmetros.

3.3.1 BSDS

Resultados Qualitativos

O BSDS aceita imagens do tipo “*soft edge map*” como os resultados do KSS e do KoSS. Para tanto, o BSDS binariza a imagem e depois a compara com as anotações humanas. A

quantidade de limiares de binarização é um parâmetro de entrada do BSDS para o cálculo das métricas. O valor adotado nestes testes foi 10.

Para avaliar a detecção de fronteiras dos métodos de integração KSS e KoSS, mostram-se alguns resultados nas Figuras 3.8, 3.9, 3.10 3.11 e 3.12. A disposição das imagens é mantida em todas as figuras: (a) mostra a imagem de entrada, (b) a referência humana, e os resultados dos métodos (c) detecção de bordas, (d) MM-Frac, (e) KSS e (f) KoSS. As imagens destes dois últimos métodos já estão binarizados em seu melhores limiares computados pelo BSDS. Cada imagem resultante tem a sua métrica *F-measure* listada logo abaixo.

Para todas as imagens mostradas é possível verificar que os resultados dos métodos KSS e KoSS são bem similares. Apenas em algumas imagens é possível notar a diferença de alguns pixels perdidos nos resultados do KSS e que não são visíveis nos resultados do KoSS. A diferença de pixels é tão pequena, que as métricas praticamente não se alteram.

Um problema em aberto do MM-Frac, que já havia sido identificado por Deng e Manjunath (2001), que é causado pelas variações suaves de tonalidade, comumente geradas por uma fonte de iluminação, foi em parte resolvido pelos métodos KSS/KoSS. Na Figura 3.6 observa-se que o KSS apaga o contorno circular dos cantos da imagem da águia ou o contorno no céu do exemplo da imagem do trem (Figura 3.7). Esta mesma funcionalidade pode ser verificada no resultado da imagem 45096, mostrada na Figura 3.8. Observe-se que os contornos da parte superior da imagem foram gerados no método de crescimento de regiões devido à uma suave luminosidade nesta parte da imagem. Estes segmentos não aparecem nos resultados dos métodos KSS e KoSS, pois o resultado do método de detecção de bordas não gera respostas nessa mesma área. É possível notar que no resultado do KSS ainda restam alguns pixels no canto superior esquerdo, que não mais existem no resultado do KoSS.

Na segunda imagem, 229086, mostrada na Figura 3.8, tem-se uma leve variação de iluminação no canto inferior esquerdo. Porém, essa região é areia, e o resultado do detector de bordas gera muita informação e, portanto, pouco confiável. Assim, infelizmente, os métodos KSS/KoSS não conseguem eliminar estes contornos, já que a lógica de ambos é retirar as informações falsas do mapa de regiões guiados pelas informações do mapa de bordas. Se o mapa de bordas não é confiável, não é possível eliminar informações.

O método de detecção de bordas é responsável por acentuar detalhes como a vara que o índio segura na imagem 101087 da Figura 3.9, ou a folha (ou inseto) na frente da cobra nos resultados da imagem 196073, ou o galho caído na neve da imagem 167062. Assim, pequenos objetos que não aparecem no resultado do método de crescimento de região, mas que respondem fortemente no método de detecção de bordas, são mantidos, ao passo que a informação excessiva gerada pelo método de detecção de bordas é atenuada e desaparece após a binarização do BSDS. Essa atenuação se deve à soma das imagens dos resultados dos

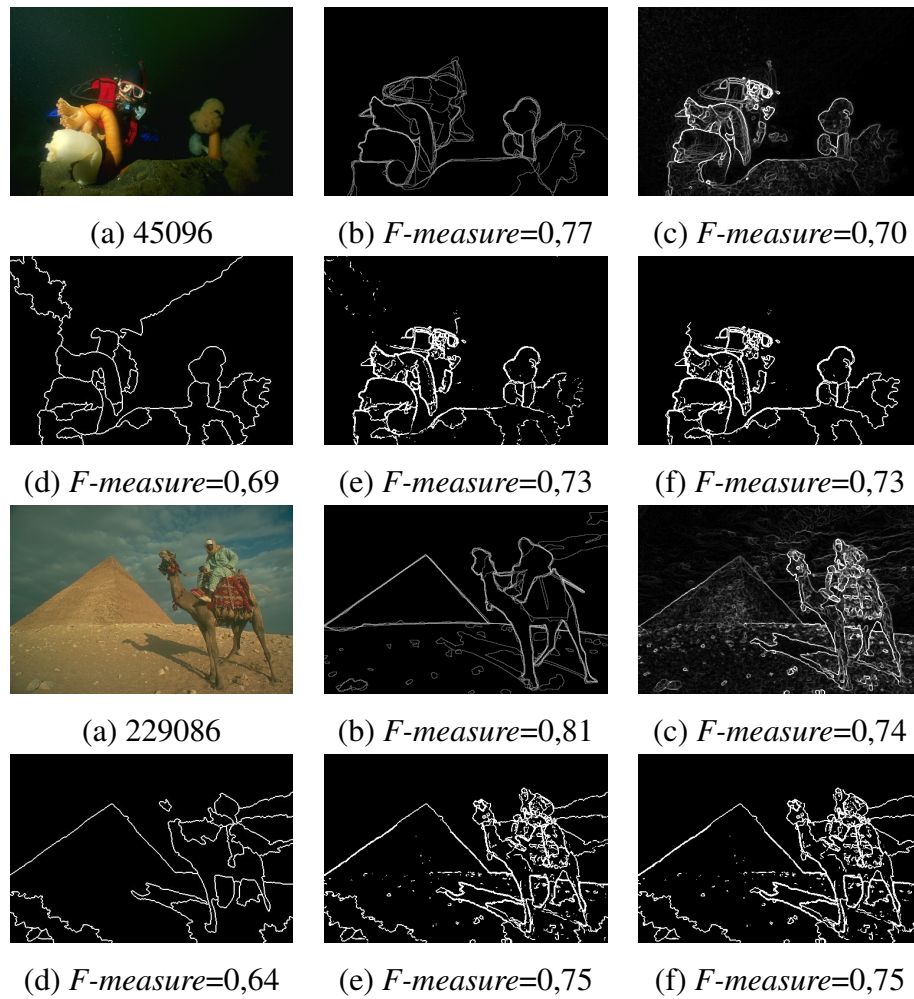


Figura 3.8: Exemplos comparativos (a) Imagem de entrada (b) Referência humana e as imagens resultados dos métodos (c) detecção de borda (d) MM-Frac (e) KSS (f) KoSS.

métodos complementares, e, logo, os pixels de borda causados pela areia (imagem 196073) ou pelas folhas das árvores (imagem 167062) não aparecem no resultado final do KSS/KoSS.

Esse realce de detalhes pode ser mais extremo, como nos exemplos da Figura 3.10, onde as linhas no chão da imagem 37073 são intensas e fazem parte do resultado da integração. Efeito similar ocorre com a imagem 227092, onde os desenhos em alto relevo do vaso fazem parte da anotação humana. Esta imagem é bem peculiar, pois apresenta figuras geométricas, quase não há texturas naturais e, além disso, a anotação humana segmentou esses elementos. Neste caso, o resultado do método de detecção de bordas é tão melhor que seu resultado sozinho já resulta no mais alto valor de $F\text{-measure}$, dentre os sistemas comparados.

Um exemplo oposto é mostrado na terceira imagem da Figura 3.10 (148026). Apesar da imagem conter figuras geométricas, como as janelas do reflexo do prédio na água, as mesmas não foram segmentadas pelos anotadores humanos. Com isso, o resultado do detector de

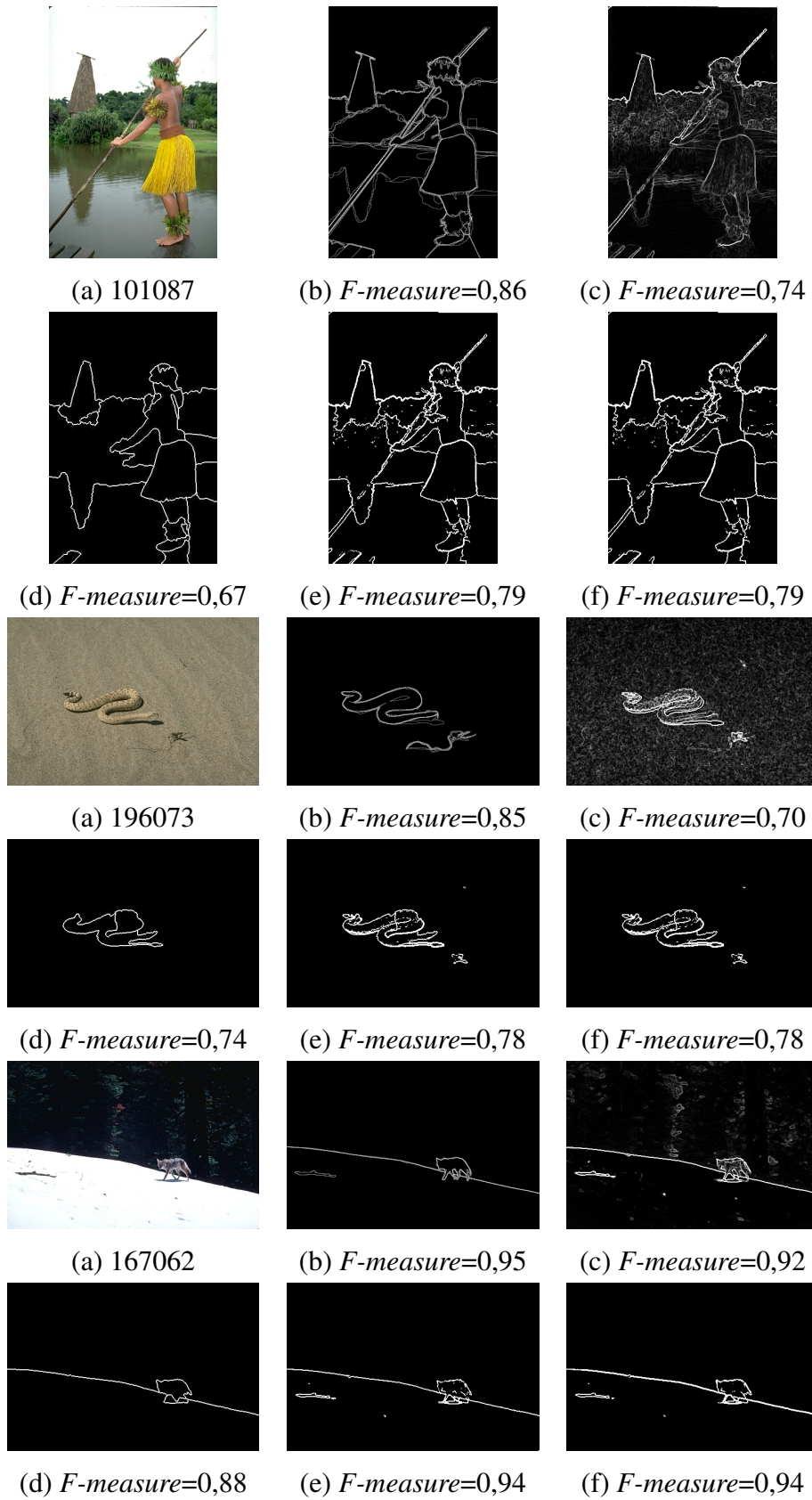


Figura 3.9: Exemplos comparativos (a) Imagem de entrada (b) Referência humana e as imagens resultados dos métodos (c) detecção de borda (d) MM-Frac (e) KSS (f) KoSS.

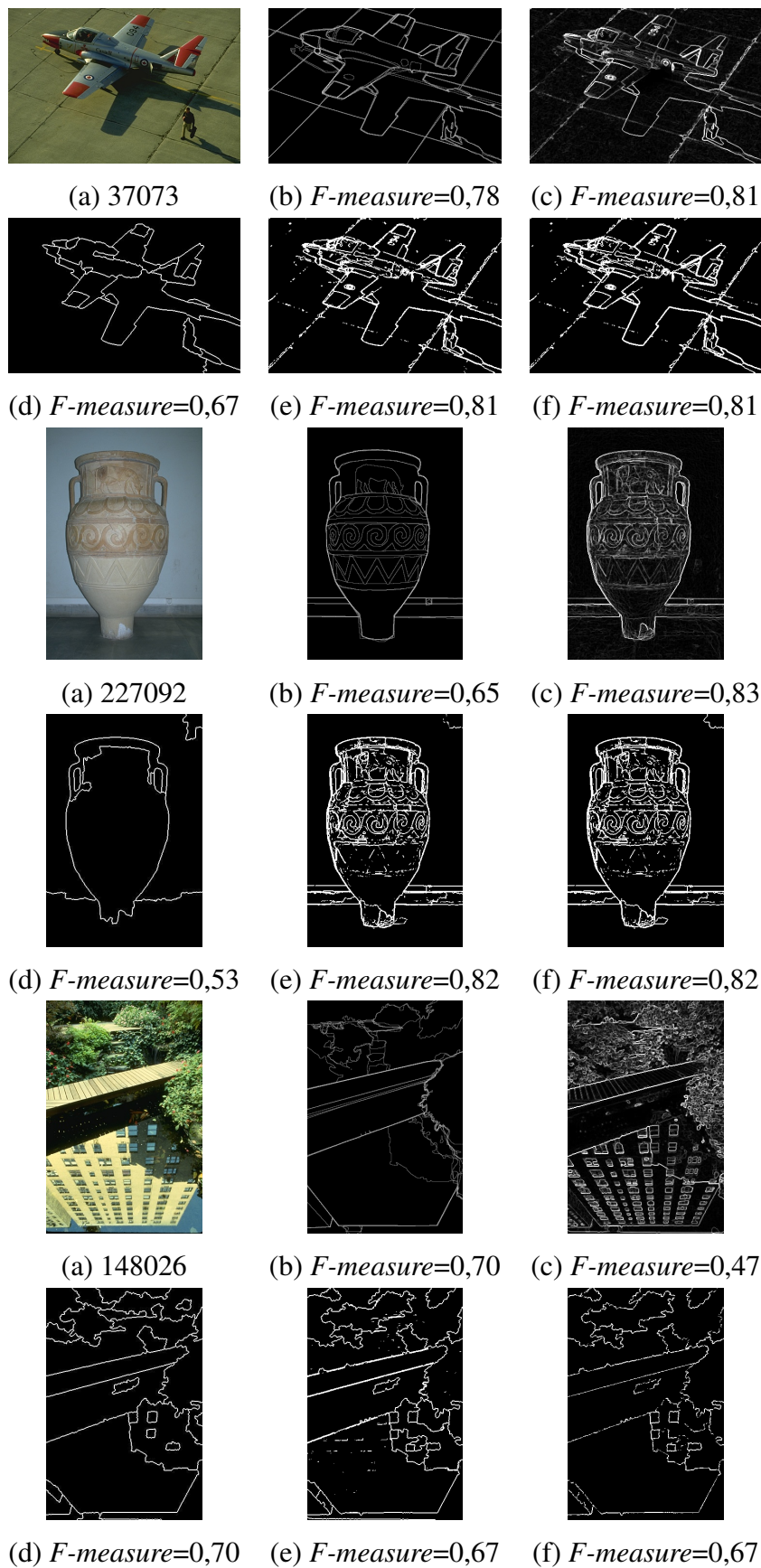


Figura 3.10: Exemplos comparativos (a) Imagem de entrada (b) Referência humana e as imagens resultados dos métodos (c) detecção de borda (d) MM-Frac (e) KSS (f) KoSS.

bordas tem valor bem menor de *F-measure*, quando comparado aos outros métodos.

Um bom exemplo de quando a soma dos contornos de região e borda gera um resultado mais próximo ao do anotador humano pode ser visto nos resultados da imagem 54082 da Figura 3.11. Nesta imagem, os anotadores humanos identificaram duas características da árvore: uma região cobrindo as folhas da árvore (como o resultado do MM-Frac) e outra com os seus vários galhos (como o resultado da detecção de bordas). A integração das duas informações, região e borda, foi importante para que o resultado final ficasse mais próximo da anotação humana.

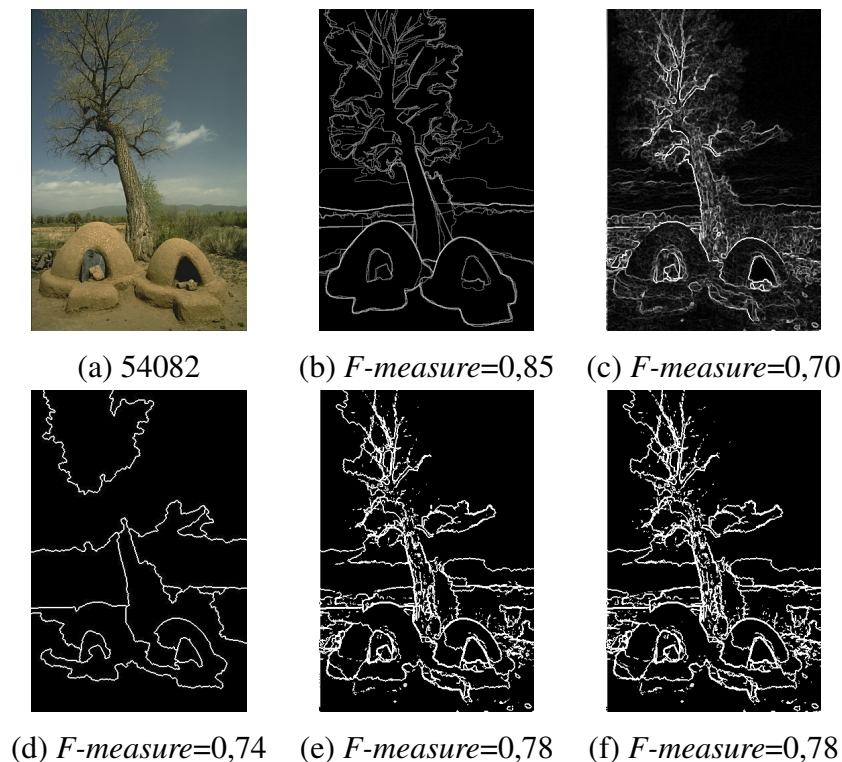


Figura 3.11: (a) Imagem de entrada (b) Referência humana e as imagens resultados dos métodos (c) detecção de borda (d) MM-Frac (e) KSS (f) KoSS.

Todos os resultados do KSS/KoSS apresentados até agora são mais similares à anotação humana, com maior valor da métrica *F-measure*, que os métodos individuais de entrada, tanto a detecção de bordas quanto o MM-Frac. As duas únicas exceções são mostradas na Figura 3.12, onde a métrica *F-measure* foi menor que as métricas de ambas as entradas.

Na imagem do bode nas pedras, 304074 da Figura 3.12, nota-se que os contornos finais do resultado KSS/KoSS apresentam falhas. Ao somar os resultados do MM-Frac e detecção de bordas, um mesmo segmento de fronteira fica com níveis diferentes de intensidade de pixels. A binarização da imagem é feita pixel-a-pixel e, com isso, os segmentos podem ficar com lacunas, causando um valor de *recall* muito baixo, influenciando assim no valor de *F-*

measure. No caso da imagem 219090 a soma dos resultados de região e borda aumentou a quantidade de pixels de bordas, e o *recall* aumentou. Relembrando que, em geral, o aumento de *recall* acarreta em diminuição de *precision*, e isto gerou um *F-measure* menor. Reitera-se que, numa base de dados de 100 imagens, só ocorreu diminuição do *F-measure* nestes dois casos.

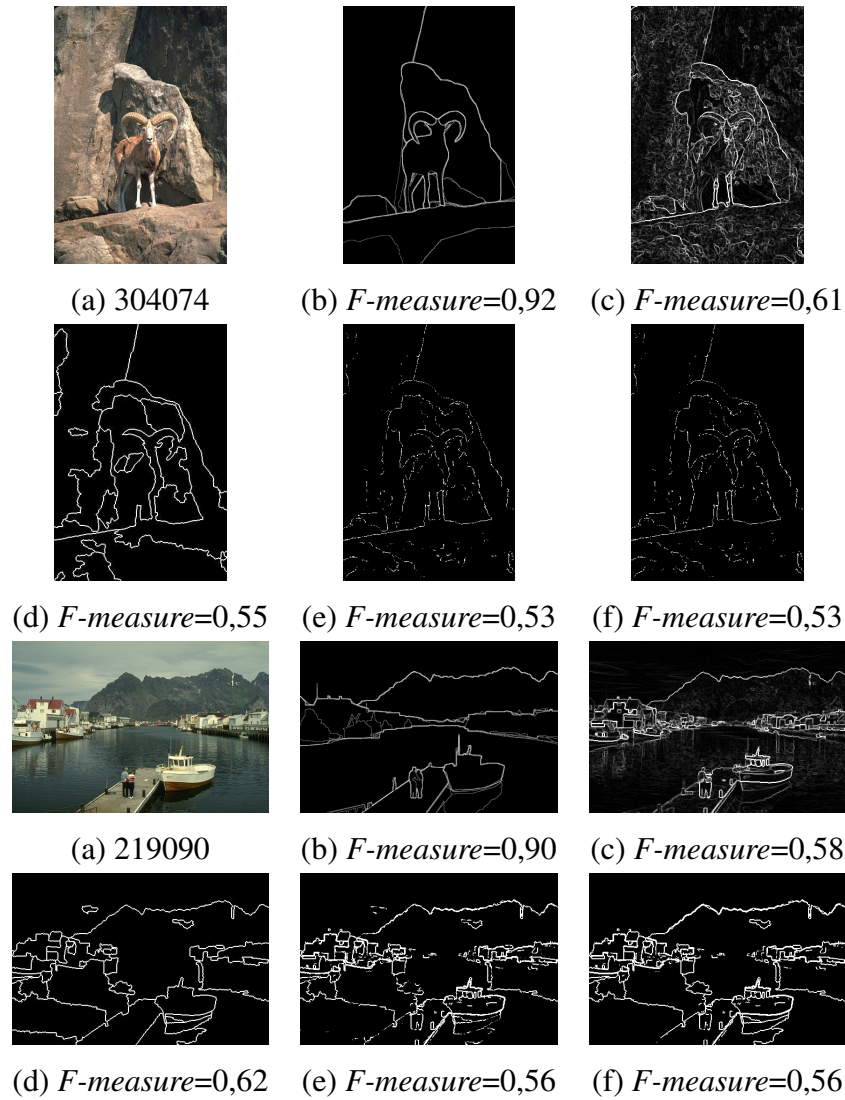


Figura 3.12: Resultados do KSS/KoSS que apresentaram *F-measure* menor que que ambos os métodos MM-Frac e detecção de borda (a) Imagem de entrada (b) Referência humana e as imagens resultados dos métodos (c) detecção de borda (d) MM-Frac (e) KSS (f) KoSS.

Resultados Quantitativos

Em termos quantitativos, as métricas *recall*, *precision* e *F-measure* de cada método, computadas pelo BSDS, são mostradas na Tabela 3.3. Conforme previsto, as métricas do KSS e

do KoSS são idênticas, visto que a diferença entre os resultados dos algoritmos resume-se a alguns pixels das imagens. A Figura 3.13 mostra o gráfico *precision* \times *recall* dos métodos de detecção de bordas (filtro de Kuwahara e gradiente morfológico), KoSS, JSEG e MM-Frac. Não se apresenta tal gráfico para o KSS porque ele é idêntico ao do método KoSS. Nas curvas de gradiente morfológico e KoSS, acrescentou-se uma marcação diferente para o ponto de maior *F-measure*. Nesta curva, quanto mais perto do ponto (1,1) melhor a métrica *F-measure*.

Tabela 3.3: Métricas *precision*, *recall* e *F-measure* calculadas pelo BSDS, comparação entre referência humana, JSEG, MM-Frac, detecção de bordas, KSS e KoSS.

	<i>Recall</i>	<i>Precision</i>	<i>F-measure</i>
JSEG	0,61	0,56	0,59
MM-Frac	0,63	0,56	0,59
Detecção de Bordas	0,65	0,49	0,56
KSS	0,69	0,54	0,61
KoSS	0,69	0,54	0,61
Referência Humana	0,70	0,89	0,79

Os métodos KSS/KoSS apresentam valores de *recall* bem próximos ao referencial humano, indicando que a quantidade de informação de fronteira é compatível, mas falha em acertá-las, pois a métrica *precision* é menor. O importante foi que a métrica *F-measure* que considera este “trade-off” entre *recall* e *precision* aumentou, e foi melhor que as métricas do MM-Frac e da detecção de bordas. Em outras palavras, a integração pós-processamento de fato contribui para melhorar a segmentação, o que é outra contribuição importante desta Tese.

Uma outra forma de visualização destas métricas é pelo histograma, onde cada ponto do gráfico representa a quantidade de imagens (eixo y) que tiveram uma determinada métrica numa faixa de valores (eixo x). Na Figura 3.14 mostra-se uma comparação dos resultados dos métodos KoSS, MM-Frac e detecção de bordas, na legenda representado pela palavra “Edge”. Pode-se observar no gráfico de *recall* (Figura 3.14b) que o método KoSS é melhor que os outros dois métodos, pois apresenta mais imagens com resultados de valores de *recall* mais altos, e logo, a sua curva está mais à direita dos demais. No gráfico da métrica *precision* (Figura 3.14a), o método KoSS possui curva quase idêntico ao do MM-Frac, apresentando apenas um pico um pouco mais elevado. Já o detector de bordas possui curva bem mais baixa que os demais, resultado esperado, pois o resultado do detector de bordas apresenta muito mais pixels de borda que o referencial humano. No gráfico de *F-measure* (Figura 3.14c), o KoSS possui curva mais à direita que os demais, mostrando melhor resultado nesta métrica.

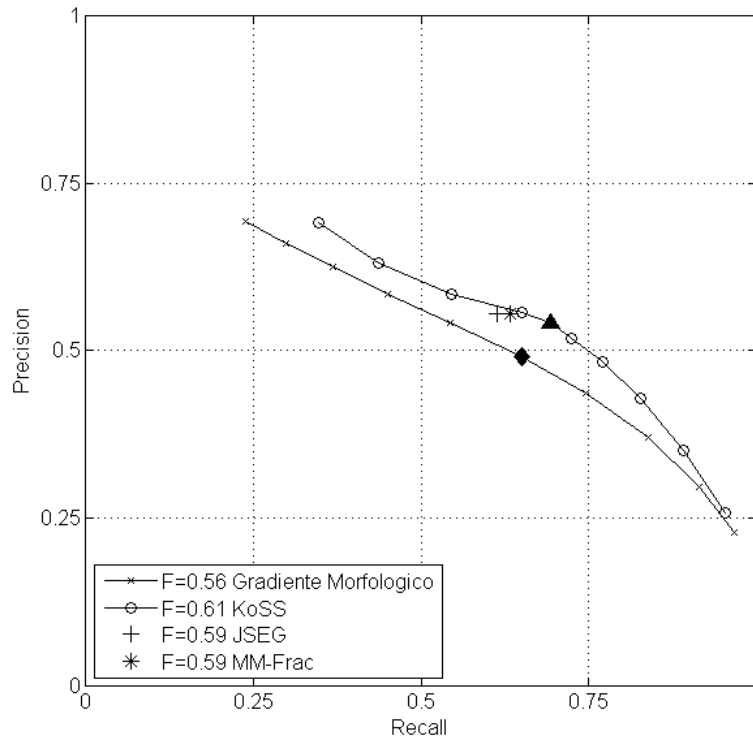


Figura 3.13: Gráfico *precision* \times *recall* dos métodos gradiente morfológico, KoSS, JSEG e MM-Frac.

Fazendo-se uma comparação par a par de métodos, com o resultado do KSS/KoSS, e considerando-se apenas a métrica *F-measure* das imagens da base de testes, tem-se que:

- o resultado do KSS/KoSS foi melhor que o resultado do MM-Frac em 82% das imagens;
- o resultado do KSS/KoSS foi melhor que o resultado da detecção de bordas em 81% das imagens;

Conforme já mostrado na Figura 3.12, há apenas dois casos nos quais o resultado do KSS/KoSS foi pior que ambas as entradas. E há 58 casos em que o resultado do KSS/KoSS apresentou métrica *F-measure* maior que o maior valor de *F-measure* do MM-Frac ou do detector de bordas.

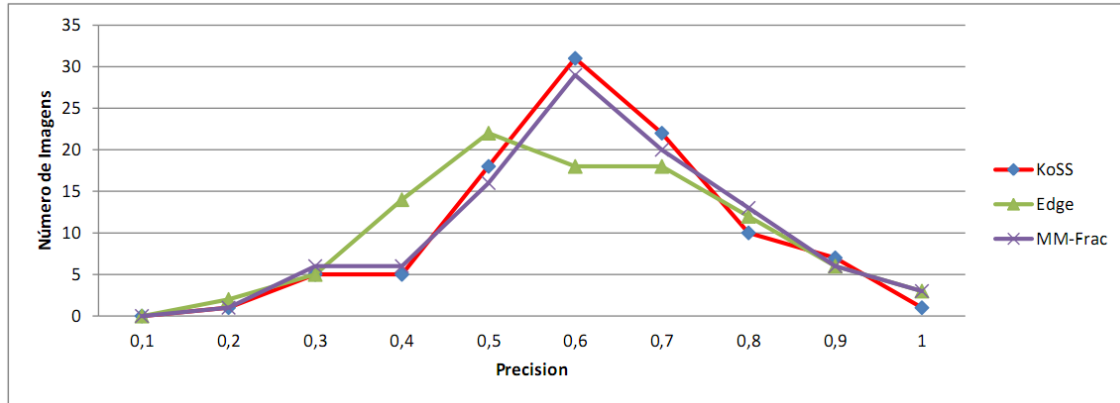
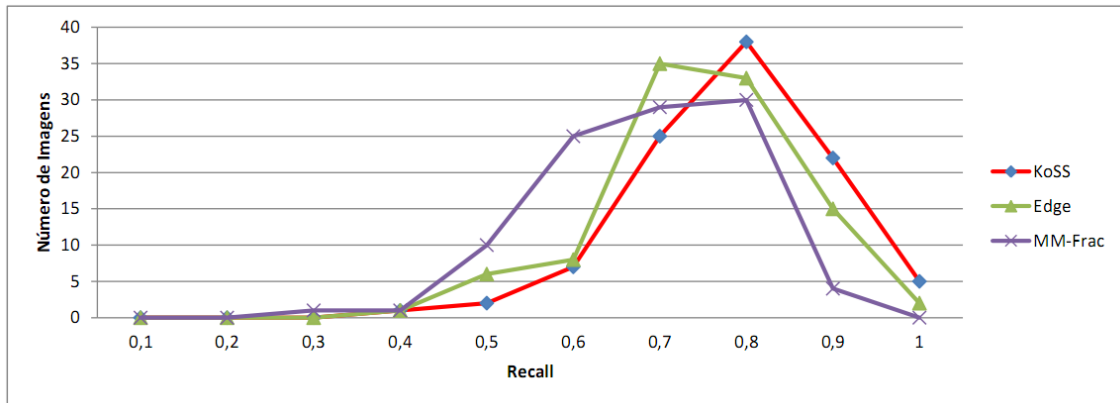
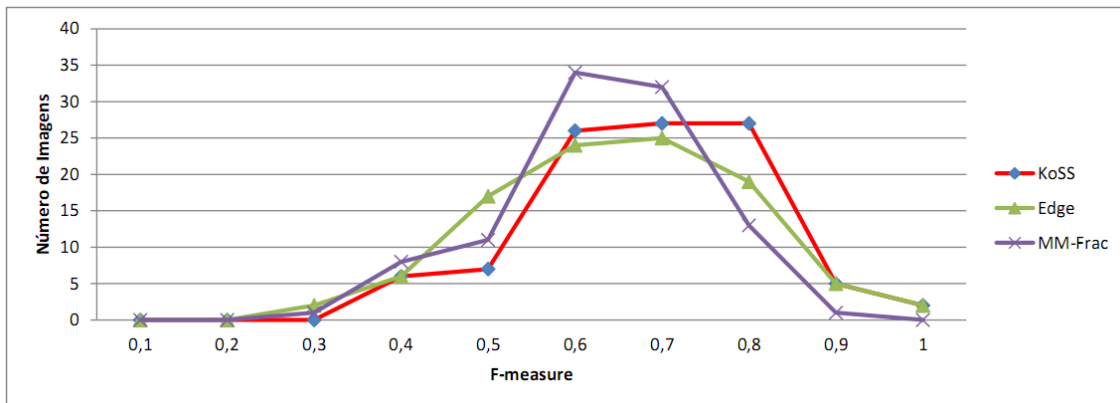
(a) Histograma da métrica *precision*.(b) Histograma da métrica *recall*.(c) Histograma da métrica *F-measure*.

Figura 3.14: Histogramas das métricas dos métodos KoSS \times MM-Frac \times detector de bordas (na legenda dos gráficos como “Edge”).

Discussão sobre o referencial humano

O objetivo do BSDS, ao criar este referencial humano, é de capturar como se processa a segmentação de indivíduos “comuns” (Martin, 2002). Para o esforço de criação do referencial humano do BSDS, 30 (trinta) indivíduos foram selecionados para a anotação humana. Foram 17 do sexo masculino e 13 do feminino. Nove indivíduos tinham formação artística e nenhum dos indivíduos tinham conhecimento associados à disciplinas de processamento de imagens e reconhecimento de padrões. Todos receberam a mesma instrução:

“Você será apresentado uma imagem. Divida a imagem em um certo número de elementos, onde os elementos representam ”coisas” ou ”partes das coisas” na cena. O número de elementos é com você, pois depende da imagem. Algo entre 2 e 30 é razoavelmente apropriado. É importante que todos os elementos tenham aproximadamente igual importância.”

Além desta instrução, foi pedido para que fosse evitado o foco em um único objeto (pelos experimentos, eles perceberam que isto acontecia geralmente com rostos humanos), que em geral gerava a exclusão de outros objetos. Isto é, pediu-se para que os indivíduos tentassem prestar igual atenção a todos os objetos na cena e não se fixar em um único objeto. Não foi transferido para os indivíduos quaisquer preocupações de pesquisadores da área de visão computacional quanto ao tratamento de sombras e reflexos.

Nota-se que o ser humano realiza a segmentação usando informações semânticas, como pode ser visto em alguns exemplos na Figura 3.15. A pele dos elefantes possui a mesma cor-textura, mas o ser humano indica uma linha divisória entre os dois animais, conforme pode ser visto na primeira linha da figura. Isto sugere que os seres humanos reconheceram os animais, e por isso distinguiram que haviam dois elefantes. Na segunda linha da Figura 3.15, os próprios pesquisadores descreveram este referencial humano como “*Hallucinated Boundaries*” (Contornos Alucinados). A imagem possui sombras que tornam obscuro o contorno dos ombros e do cabelo, porém, estes contornos foram anotados por quatro dos indivíduos de testes, sugerindo que os indivíduos utilizaram conhecimento prévio da forma de pessoas.

Algumas imagens apresentam um nível de dificuldade acima dos demais, até mesmo para os seres humanos, como é o caso mostrado na Figura 3.16. Nesta imagem, é um desafio definir quantas “coisas” existem e quais são os seus limites. Além da textura ser complexa, acredita-se que a imagem não apresenta elementos comuns ao dia-a-dia da maioria das pessoas, e, com isso, não há como utilizar o conhecimento prévio. O valor de $F\text{-measure}=0,62$ é um valor baixo para o padrão humano.

Na Figura 3.17, o desafio é compreender porque os indivíduos de testes definiram os contornos em alguns tentáculos e em outros não. Nota-se que não há muita coerência mesmo

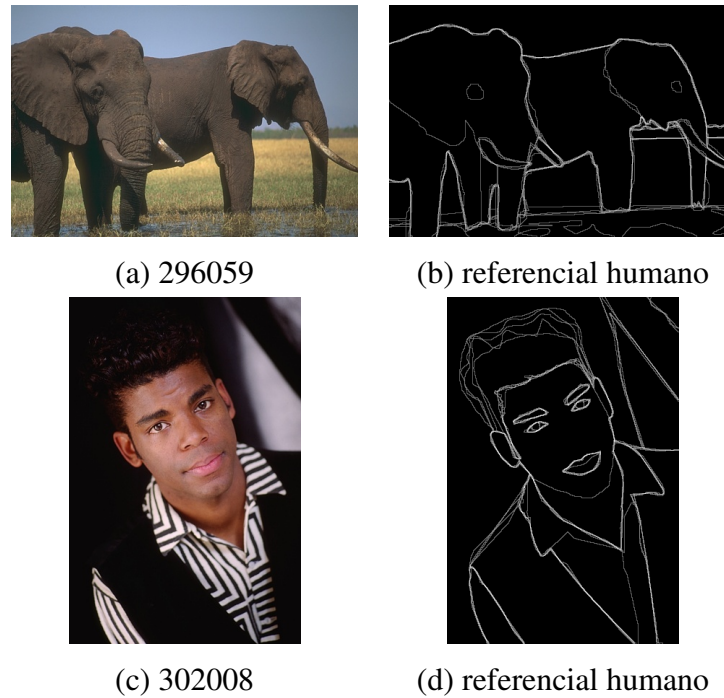


Figura 3.15: Imagens 296059 e 302008 com seus referenciais humanos.

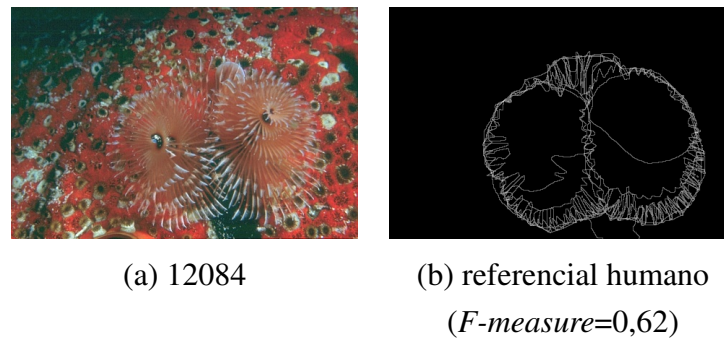


Figura 3.16: Imagem 12084 com seus referencial humano.

entre os seres humanos, dado o seu baixo valor de $F\text{-measure}$.

Já foram mostradas imagens de difícil reconhecimento devido à camuflagem natural de alguns animais e seu meio ambiente, tal qual a cobra nas areias do deserto. Um outro exemplo é mostrado na Figura 3.18, um canguru no meio de galhos secos, apresentando baixo valor de $F\text{-measure}$ para o padrão humano.

A proposta desta Tese não abrange os problemas citados nesta seção, pois necessitaria de uma estratégia, tal qual um ser humano, para armazenamento e acesso às informações do conhecimento prévio de objetos e suas formas.

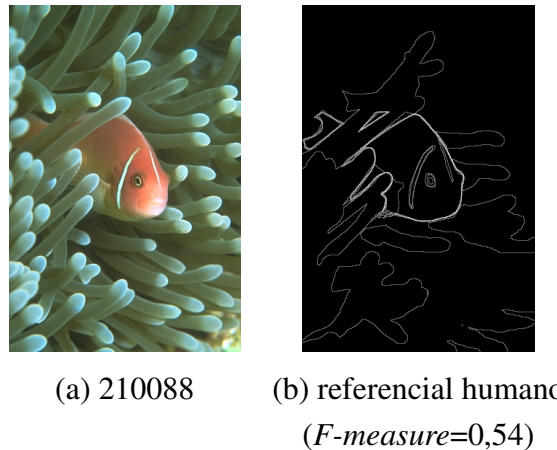


Figura 3.17: Imagens 210088 com seu referencial humano.

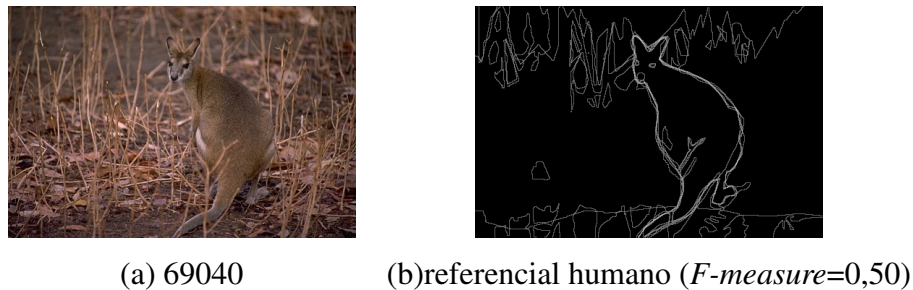


Figura 3.18: Imagem 69040 com seu referencial humano.

3.3.2 LHI

De acordo com Estrada e Jepson (2009), há duas escolas com relação à avaliação de algoritmos de visão computacional em geral. Uma escola de pensamento que sustenta que os algoritmos podem ser avaliados comparando-se o quão próximos os seus resultados estão da anotação humana, e outra escola que sustenta que algoritmos de visão devem ser avaliados no contexto da tarefa específica. Nesta segunda linha, a qualidade da segmentação de imagens e outras atividades de baixo nível (*low level task*) se traduz em medir a sua contribuição para o sucesso da tarefa de nível superior (por exemplo, o reconhecimento de objetos).

Nesta Tese, é seguida a primeira escola, isto é, considera-se uma comparação com o referencial humano. Há vários bancos de dados nesta linha. No entanto, desconsiderando os bancos de dados de imagens médicas, mosaicos de texturas, imagens não-coloridas, imagens astronômicas, imagens de faces humanas, o único banco de dados de imagens de cenas naturais coloridas cujo objetivo é segmentar regiões é o BSDS (Estrada e Jepson, 2009).

Assim, a maioria dos bancos de dados de cenas naturais coloridas trata o problema de classificação de imagens ou de reconhecimentos de objetos. Com isso, o referencial com-

parativo das anotações humanas incorpora as considerações semânticas que estão além do escopo desta Tese. Nestes casos, o referencial humano de banco de dados para classificação de imagens ou de reconhecimentos de objetos sempre apresenta uma quantidade de regiões menor que a do BSDS.

O banco de dados PASCAL VOC (*Visual Object Classes Challenge*) incluiu uma competição específica de segmentação desde a sua versão de 2007, (Everingham et al., 2007, 2008, 2009, 2010, 2011). Na Figura 3.19 mostram-se alguns exemplos que não condizem com o resultado de um algoritmo de segmentação de regiões. Na primeira linha desta figura, a imagem possui uma casa e vegetação à esquerda e à direita. Porém, o *benchmark* do banco de dados só segmenta uma pessoa ao fundo, um arbusto ao fundo e uma única fronteira para o vaso com a planta que está à frente da casa nesta imagem. Da mesma forma, nas outras imagens, há muito mais regiões nas cenas, mas o referencial se limita apenas à identificação dos contornos de elementos existentes no conjunto limitado de 20 (vinte) diferentes objetos.

Característica parecida é encontrada no *Microsoft Research in Cambridge Object Recognition Database* (MSRC) (Shotton et al., 2006), que possui 591 imagens contando com 23 classes de objetos. Em muitas imagens, a anotação humana traça apenas o contorno ao redor de um objeto, lançando todo o restante da imagem como pano de fundo, dificuldade discutida em Rao e colegas (Rao et al., 2009). Artigos da equipe mostram resultados como na Figura 3.20, onde o importante não é traçar o contorno do objeto, e sim, reconhecer a classe do(s) objeto(s) em evidência, e o contorno se resume a um retângulo em torno do objeto.

Um esforço interessante está sendo realizado por Russel e colegas (Russell et al., 2008), com o objetivo de criar coleções de referencial humano. O projeto usa a ferramenta *online* “LabelME web” disponibilizando ao público a criação de segmentações e “nomeações” (atribuição de uma palavra a cada elemento segmentado). As imagens são de cenas naturais, mas as segmentações são definidas por polígonos em torno dos objetos. Assim, embora as vantagens deste banco de dados sejam o seu tamanho e o grande volume de anotação humana, não há ferramenta para se desenhar detalhadamente os contornos das segmentações.

O banco de dados de imagens que mais se aproxima do propósito deste trabalho, além do BSDS, é o “*Lotus Hill Research Institute*” (LHI) (Yao et al., 2007). Este projeto durou dois anos, contando com a colaboração de uma equipe de 23 pessoas em tempo integral, de duas instituições, a “*Lotus Hill Research Institute*” da China, e o “*UCLA Center for Image and Vision Sciences*” (CIVS) dos EUA. O projeto fornece um ambiente de anotação para o usuário humano com três níveis: nível de cena (classificação de cenas e descrição de atividades, dentre outros), o nível de objeto (tais como a decomposição hierárquica de objetos em partes, anotação semântica e *sketching*) e de nível médio (contorno de objetos e complemento de oclusão e outras facilidades). O objetivo final do projeto é fornecer imagens e referencial

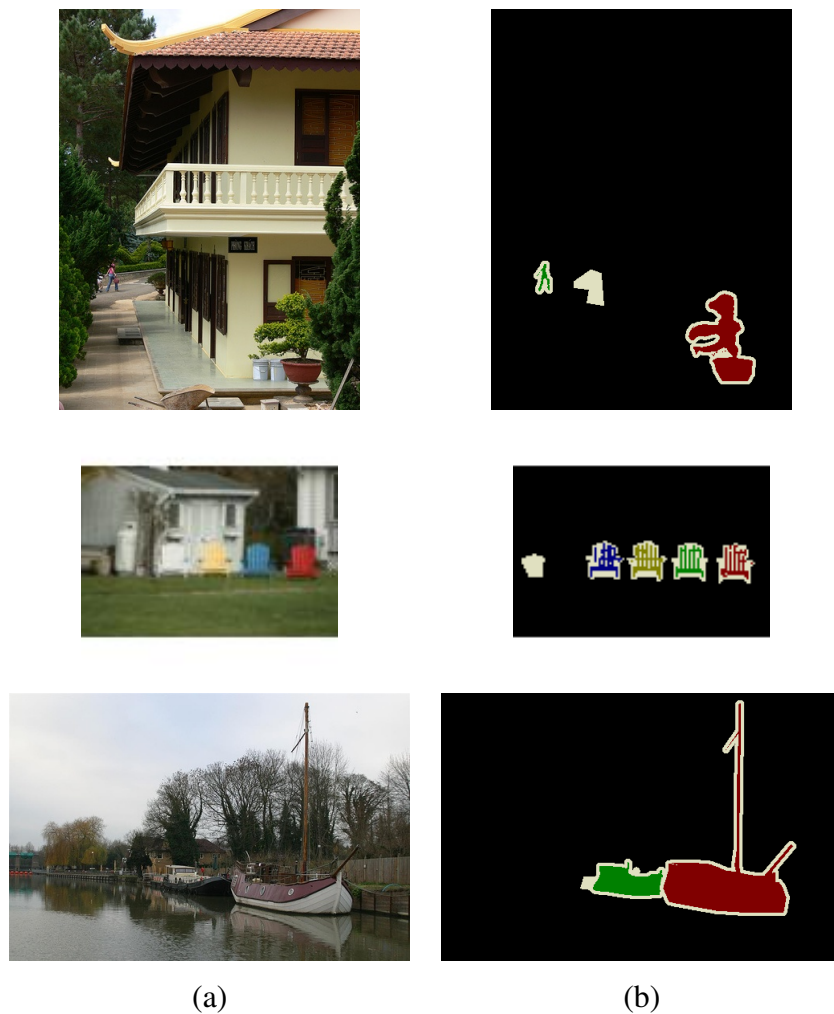


Figura 3.19: Exemplos do PASCAL VOC, onde a coluna esquerda (a) mostra as imagens de testes do banco de dados e a coluna direita (b) mostra o referencial humano das imagens à sua esquerda.

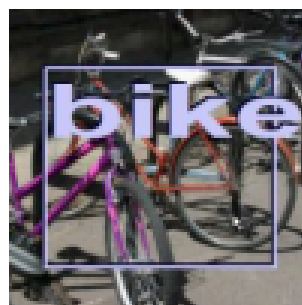


Figura 3.20: Resultado desejado em pesquisas utilizando o MSRC.

para o nível mais alto, que é o de cena. No nível mais baixo, existe um referencial de contorno de objetos e não de regiões. Mas, pelo menos, os referenciais humanos fornecidos possuem um contorno mais preciso do que os dos bancos de dados citados anteriormente.

A finalidade destes testes no LHI é para validar a arquitetura proposta em outro banco de dados de imagens, que apresenta resoluções e tamanhos diferentes dos encontrados no BSDS. Porém, o LHI apresenta como *benchmark*, imagens com bem menos regiões do que o esperado, e, logo, são feitos apenas testes qualitativos e não quantitativos.

Processamento de imagens após resultado do KoSS

O sistema associado ao BSDS fornece uma facilidade de aceitar imagens em escala de cinza como resultados de segmentação e um parâmetro de entrada, número inteiro maior que um, que indica a quantidade de limiares para a binarização destas imagens. Já o LHI não possui esta facilidade. Assim, acrescentou-se uma fase à saída do KoSS, de binarização e retirada de pequenas áreas conectadas, descrita no código do Algoritmo 4.

Algorithm 4 Processamento após KoSS

```
1: Parametros de entrada: NomeArquivoEntrada, NomeArquivoSaida, Tamanho
2: Img = imread(NomeArquivoEntrada);
3: [x, y] = size(Img);
4: I2 = im2bw(Img);
5: I3 = bwareaopen(I2, Tamanho*x*y);
6: imwrite(I3, NomeArquivoSaida, 'png');
```

Este algoritmo foi desenvolvido em Matlab, e para cada imagem, no passo 1, recebe três parâmetros de entrada: o nome do arquivo de entrada (**NomeArquivoEntrada**), o nome do arquivo de saída (**NomeArquivoSaida**) e o tamanho (**Tamanho**).

Após a leitura do arquivo de imagem (passo 2) e a extração da altura e largura da imagem (passo 3), chama-se a função **ImagemPretoBranco = im2bw(ImagemEscalaCinza)** (passo 4). Esta função do Matlab converte uma imagem em escala de cinza passada como parâmetro para uma imagem binária (preto e branco), usando o limiar calculado pelo método de Otsu (Otsu, 1979).

No passo 5, utiliza-se a função **NovaImagem = bwareaopen(ImagemPretoBranco, TamanhoDaArea)**, que remove da imagem binária passada como parâmetro, todos os componentes conectados que têm menos pixels que o parâmetro **TamanhoDaArea** (que é o resultado do parâmetro **Tamanho** multiplicado pela largura e altura da imagem), produzindo uma imagem binária como saída. A conectividade padrão da área é 8 (oito) para duas dimensões. O parâmetro de tamanho (**Tamanho**) passado para todas as imagens foi sempre de 0,05%. O propósito deste parâmetro ser bem baixo é para retirar apenas regiões pequenas.

Resultados Qualitativos

A base de dados LHI é bem vasta. No entanto, a maior parte são imagens de vídeo para sistemas de vigilância, imagens de rostos para sistemas de reconhecimento de faces, documentos digitalizados em inglês e chinês para sistemas de OCR e afins, e imagens com um único objeto, muitas vezes com fundo artificial, para sistemas de reconhecimento de objetos e imagens de ambientes fechados (*indoor*).

Neste trabalho, foram tomadas 40 (quarenta) imagens para teste, das seguintes classificações do LHI:

1. oito imagens de vistas aéreas;
2. oito imagens de animais;
3. oito imagens de pessoas praticando esportes;
4. oito imagens de cenas naturais.
5. oito imagens de edificações e rodovias.

O tipo de imagem que mais difere do estilo de imagens do BSDS é a imagem aérea. Apresentam-se nas Figuras 3.21, 3.22, 3.23 e 3.24, resultados de quatro das oito imagens, que caracterizam bem como foi a resposta geral da arquitetura para este tipo de imagem. Nestas figuras, a ordem de apresentação é (a) a imagem original, (b) o referencial humano, seguindo-se os resultados dos métodos (c) JSEG, (d) I-Frac (e) MM-Frac (f) detecção de bordas (g) Koss e (h) KoSS com o pós-processamento.

Infelizmente, para este tipo de imagem, com muita informação e com o referencial humano indicando um mínimo de contornos, o resultado do método de detecção de bordas retorna tanta borda que o KoSS não confia em sua informação. Nas Figuras 3.21, 3.22 e 3.23, pode-se notar que o JSEG segmenta mais do que as saídas I-Frac e MM-Frac. Analisando apenas qualitativamente, quase não há diferenças entre os resultados do I-Frac e do MM-Frac. Na Figura 3.24, o referencial humano faz uma espécie de loteamento, e não segmenta, por exemplo, a pista de corrida oval que existe na parte central/inferior da imagem. Na Figura 3.25, mostra-se o resultado final sobreposta (em branco) à imagem original, facilitando a visualização de que o resultado do KoSS com pós-processamento delimitou a pista de corrida.

Nestas imagens áreas, o anotador humano foi orientado a identificar os seguintes objetos: telhados de edifícios, áreas de estacionamento (carros ou barcos), carros/conjunto de

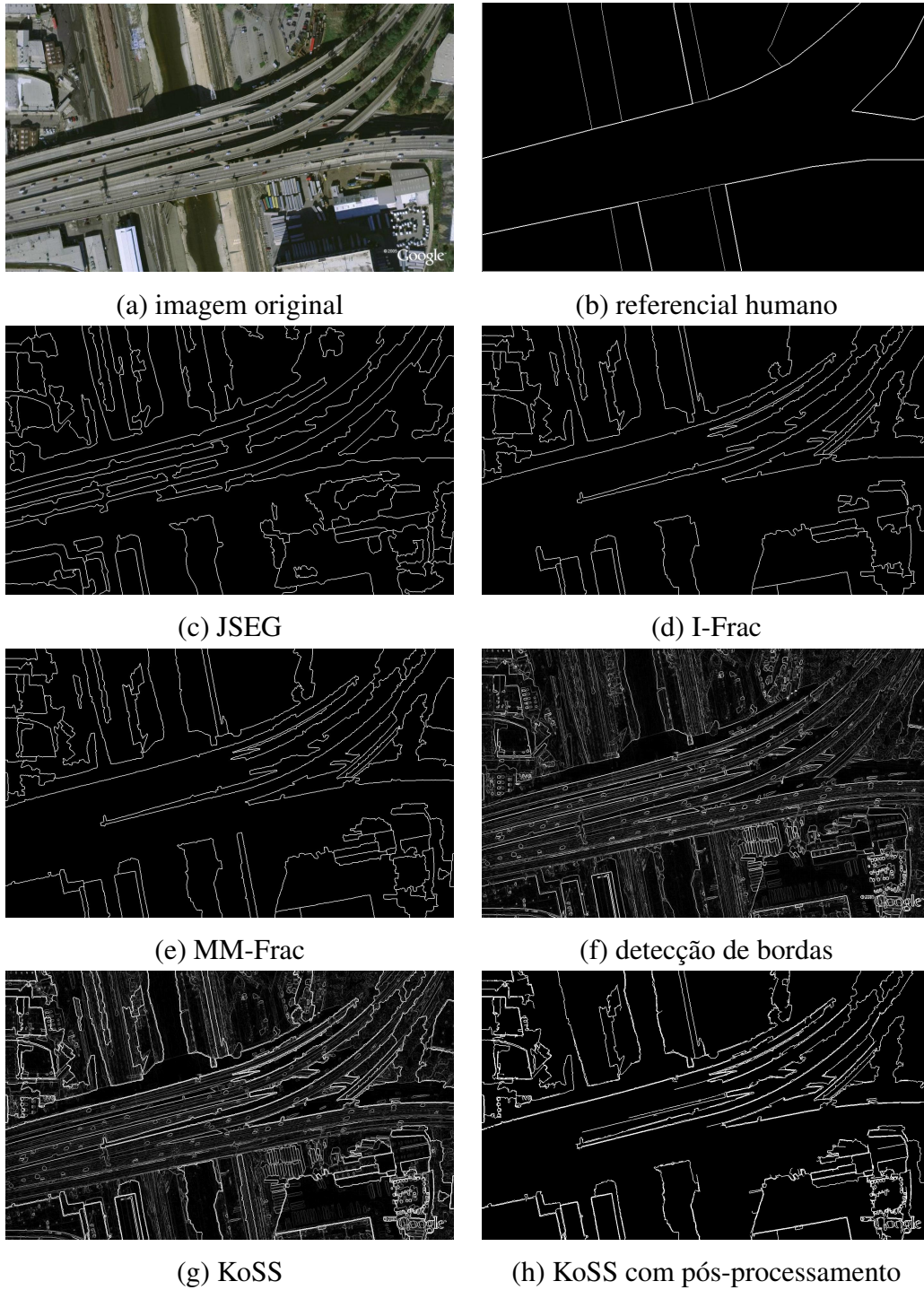


Figura 3.21: Resultados para a imagem do LHI (tamanho 1400×838).



Figura 3.22: Resultados para a imagem do LHI (tamanho 1400×838).

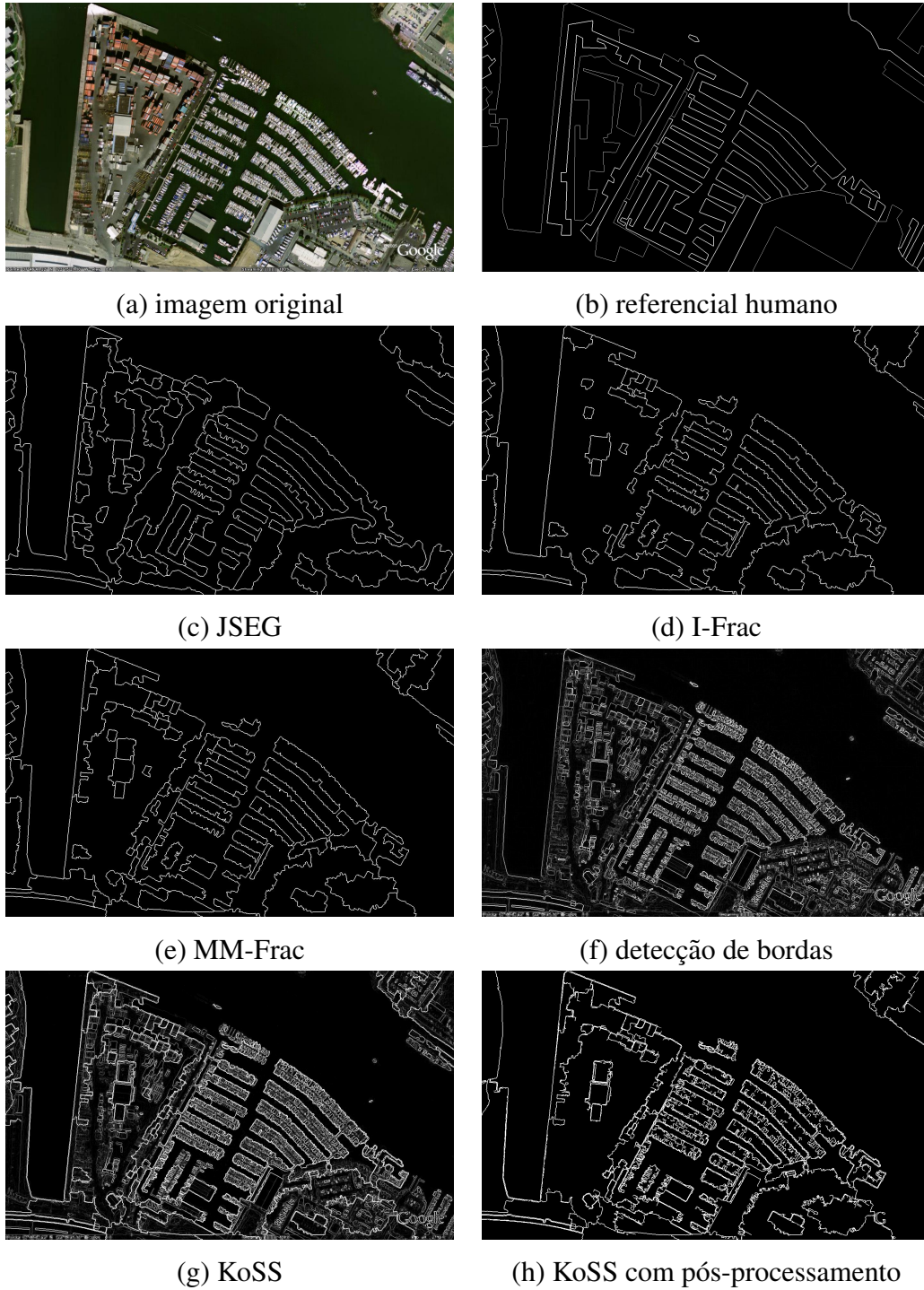


Figura 3.23: Resultados para a imagem do LHI (tamanho 1400×838).

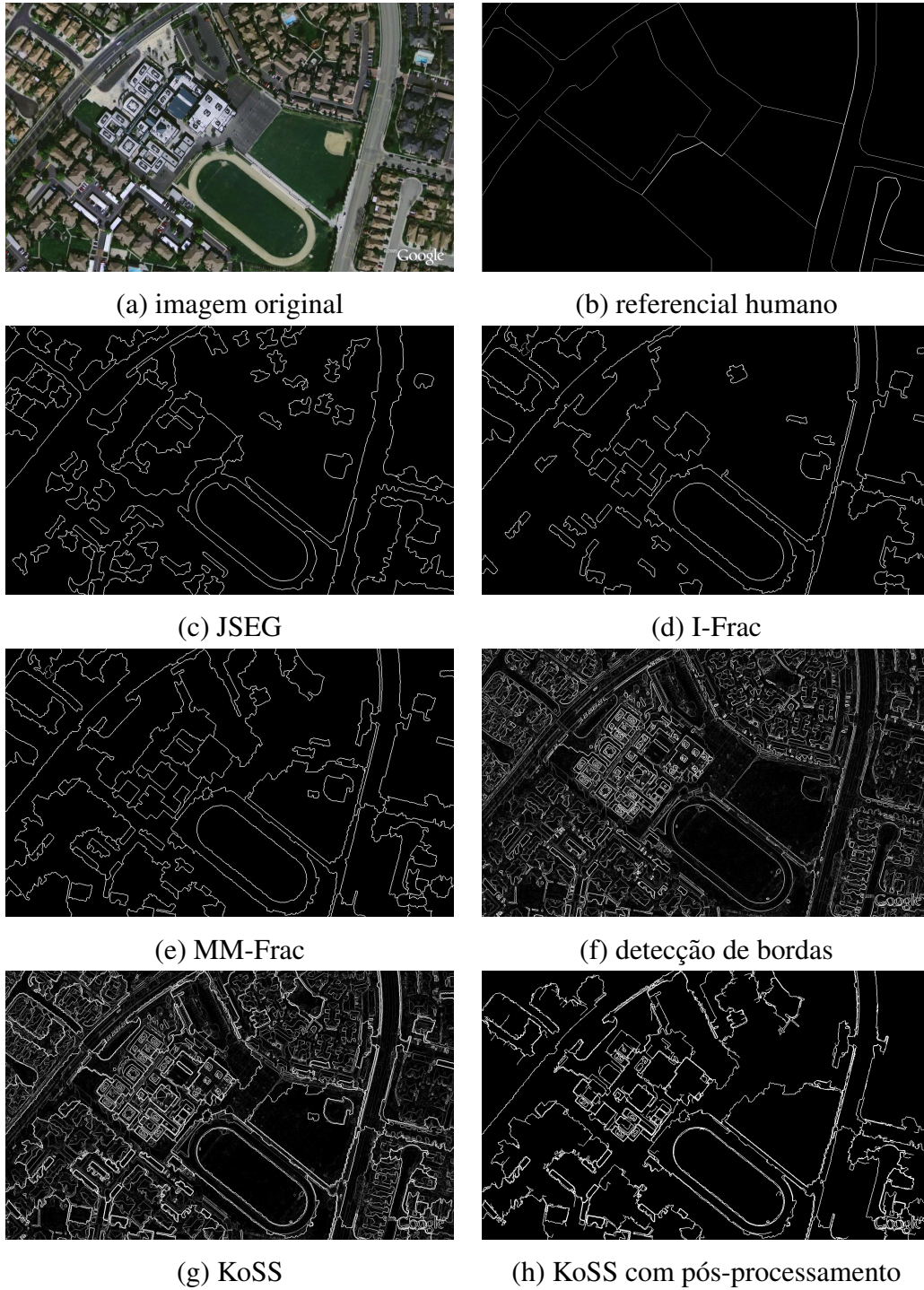


Figura 3.24: Resultados para a imagem do LHI (tamanho 1400×838).

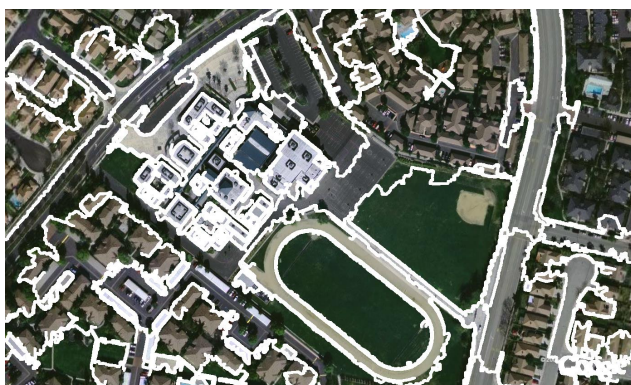


Figura 3.25: Exemplos do LHI, imagens resultante sobrepostas à imagem original.

carros e superfícies de estrada, pois estes são os objetos a serem identificados na etapa de reconhecimento. Pode-se afirmar que este referencial não é compatível com o propósito de um sistema de segmentação de regiões de cenas naturais.

Alguns resultados com outros tipos de de imagem são mostrados nas Figuras 3.26 e 3.27. A ordem das imagens segue a ordem de apresentação: (a) é a imagem original, (b) o referencial humano, seguindo-se os resultados dos métodos (c) JSEG, (d) MM-Frac (e) detecção de bordas (f) Koss e (g) KoSS com o pós-processamento.

Nas quatro primeiras linhas da Figura 3.26 pode-se verificar que as imagens do referencial humano (coluna b) identificam apenas o contorno do elemento que faz parte do conjunto de objetos que devem ser reconhecidos por sistemas de classificação, logo, todo o restante da imagem é desprezado pelo referencial humano. Note-se que na terceira imagem a borboleta possui tamanho quase igual ao da flor, mas apenas o inseto foi definido no referencial. Já de antemão, sabe-se que o resultado final da arquitetura aqui proposta não será igual ao referencial, provavelmente contendo muito mais segmentos de fronteiras.

Conforme esperado, o resultado do MM-Frac apresenta uma quantidade menor de regiões que o resultado do JSEG. Alguns detalhes do resultado da detecção de bordas são repassados ao resultado do KoSS, como os detalhes das asas das duas borboletas (terceira e quarta linhas da Figura 3.26) e a ponta do bico do flamingo (segunda linha da Figura 3.26), que não havia sido detectado pelo MM-Frac.

De acordo com a concepção da proposta deste trabalho, sempre haverá divisões de regiões, mesmo que faça parte do segundo plano. Por exemplo, na primeira linha, a imagem do gato, a definição do contorno da vegetação deve ser feita, pois há uma diferença de cor-textura desta área com a imagem em volta. Da mesma forma, o sistema irá detectar o contorno das folhas nas duas imagens de borboletas. Um bom resultado é que o sistema foi capaz de identificar o contorno dos elementos definidos no referencial humano, mesmo

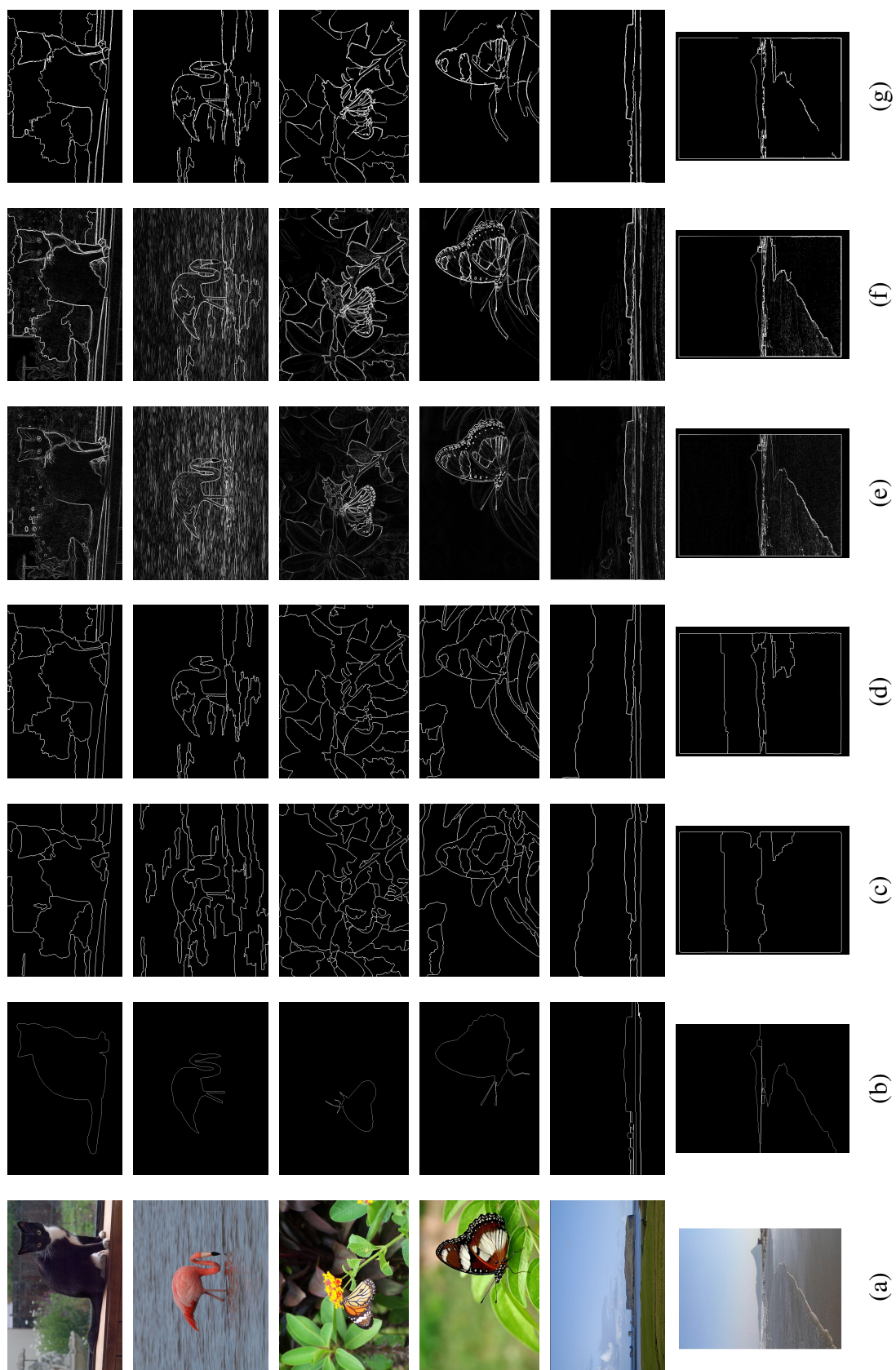


Figura 3.26: (a) Imagem original (b) referencial humano e os resultados dos métodos (c) JSEG (d) MM-Frac (e) detecção de bordas (f) KoSS (g) KoSS com pós-processamento.

tendo respondido com mais segmentações.

Alguns contornos falsos foram eliminados do MM-Frac, como as regiões da parte superior esquerda da segunda borboleta (quarta linha). O mesmo ocorre nas duas últimas linhas da Figura 3.26, onde há uma linha divisória no céu em cada uma das imagens, que também foram eliminadas pela arquitetura proposta.

Outros contornos não foram eliminados, pois a informação do resultado do detector de bordas foi muita intensa, como o movimento da água no lago onde está o flamingo. O resultado do detector de bordas preencheu a imagem inteira, e não permitiu que o KoSS pudesse eliminar as bordas falsas que o MM-Frac gerou na água.

Na última linha da Figura 3.26 está bem nítida a diferença de segmentação entre o resultado do JSEG e o resultado do MM-Frac. O método JSEG não encontrou a linha divisória das montanhas ao fundo, e foi graças ao método de detecção de bordas que a divisão entre o mar e a areia foi encontrada. O resultado final das últimas duas linhas da Figura 3.26 ficou qualitativamente bem parecido com o resultado do referencial humano.

Na primeira linha da Figura 3.27, o resultado final também ficou próximo ao referencial humano. Ocorreram algumas diferenças, como a espuma na onda que foi segmentada pelo sistema e não pelo ser humano, e o guarda-sol teve cada parte, cada cor, segmentada pelo sistema, ao passo que o ser humano compreendeu o guarda-sol como um único elemento.

Na segunda e terceira linhas, os resultados do MM-Frac detectaram mais regiões que o método JSEG. Em ambas as imagens, a vegetação composta pelas árvores apresenta mais subdivisões que o referencial humano. Na segunda linha, o pós-processamento simplificado acabou por retirar os carros da rodovia, que estavam presentes no resultado do KoSS e que estavam no referencial humano, o mesmo ocorrendo com o contorno da bola vermelha da terceira linha.

Na quarta linha da Figura 3.27, tem-se uma paisagem natural. Observa-se que a anotação humana foi bem simplista, ao traçar apenas duas fronteiras. Os resultados intermediários e final segmentam a espuma como uma região diferente e não compreendem o pier como uma região única e sim as suas partes, o que parecem ser cubos de concreto. Para conseguir abarcar a granularidade desta região de cubos, o tamanho da janela a ser utilizado deve ser tão grande quanto um dos elementos. Da mesma forma, devido à grossa granularidade, há muito mais regiões na área das rochas nos resultados (na parte esquerda da imagem) na penúltima linha da Figura 3.27.

Na última linha da Figura 3.27, tem-se uma paisagem de construções. A anotação humana faz apenas o contorno externo de todos os prédios, ao passo que o resultado final

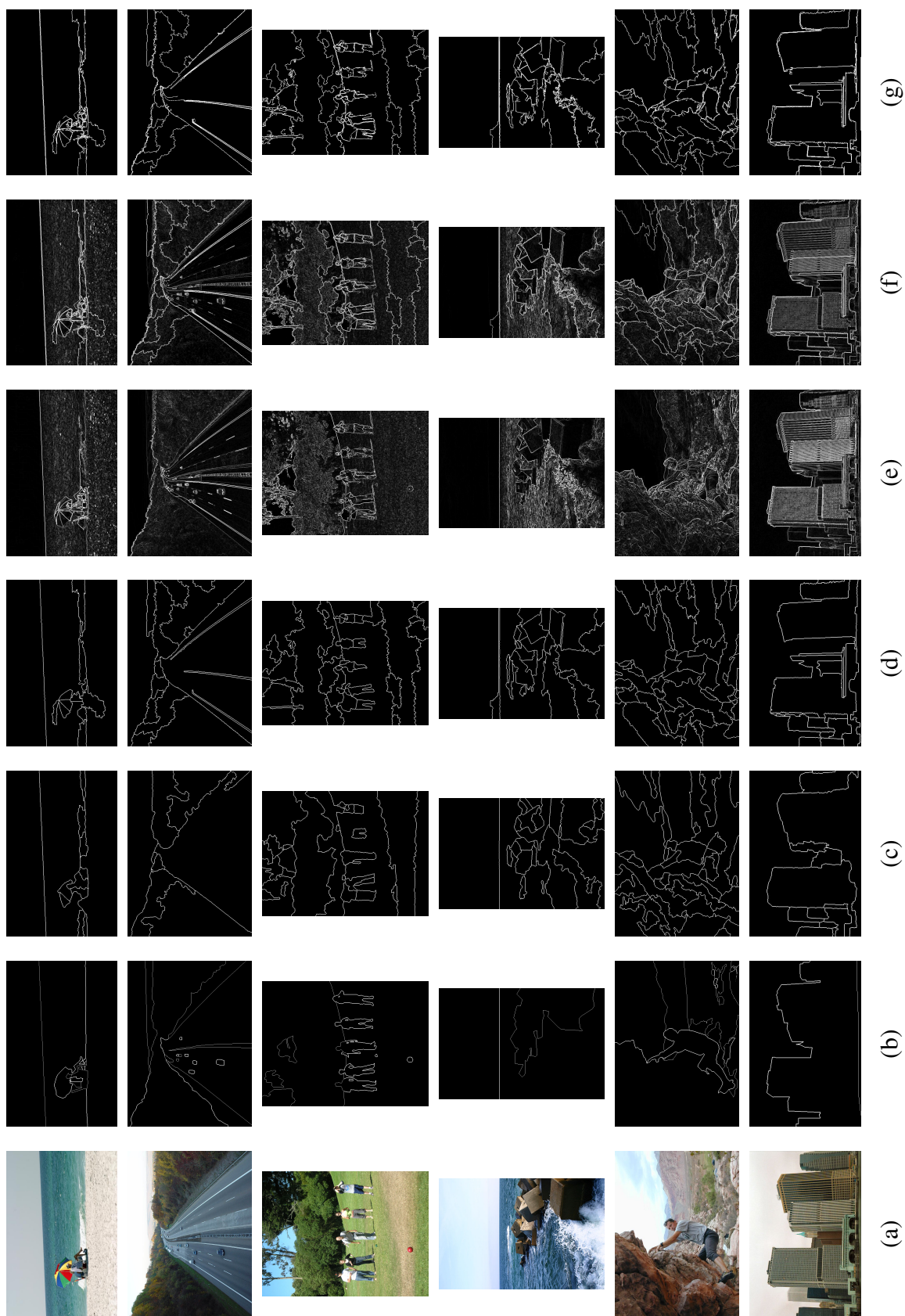


Figura 3.27: (a) Imagem original (b) referencial humano e os resultados dos métodos (c) JSEG (d) MM-Frac (e) detecção de bordas (f) KoSS (g) KoSS com pós-processamento.

identifica cada um dos prédios. A concepção do sistema é essa, de identificar cada região que contenha uma textura diferente. Por isso, o resultado final diferirá do esperado a partir do referencial humano. Nota-se como o conjunto pré-definido de classes direciona o grau de refinamento e detalhamento da anotação humana. A Figura 3.28 mostra a imagem final resultante da arquitetura (em vermelho) sobreposta à imagem original, para facilitar a visualização da detecção dos contornos de cada prédio.



Figura 3.28: Exemplos do LHI, imagem resultante sobreposta às imagem original.

3.4 Considerações Finais

Este capítulo fez uma investigação de como integrar o resultado de dois métodos baseados em diferentes abordagens, detecção de bordas e orientado a regiões, para o problema de segmentação de imagens coloridas. Embora tenham sido desenvolvidos dois algoritmos, KSS e KoSS, ambos possuem a mesma estratégia, que é eliminar informação incorreta do resultado de um método baseado no resultado do outro.

Consideram-se dois métodos de segmentação: por um lado tem-se o gradiente morfológico, que é um detector de bordas capaz de gerar altos valores de *F-measure* para determinadas imagens, enquanto falha totalmente em outras. Bons resultados são encontrados quando a imagem de entrada tem apenas elementos com contornos bem definidos, e não há texturas que provoquem um resultado “ruidoso” (com muita informação). A causa desse comportamento é da natureza do método, que se fia apenas na diferença de contraste da informação da vizinhança. Por outro lado, tem-se o MM-Frac, que apresenta resultados melhores quando a imagem apresenta muitas texturas de granularidades diferentes. Porém, apresenta problemas como contornos falsos, (por exemplo, causada pela suave gradação de iluminação), além de não detectar muitos dos pequenos componentes finos das imagens.

A estratégia de integração diminui duas informações: contornos falsos do mapa de regiões, baseado na informação do mapa de bordas, e a informação excessiva do mapa de bordas, baseado no mapa de regiões. Ao mesmo tempo, enfatizam-se: as bordas que coincidem nos diferentes métodos, e mantêm-se informações de bordas, quando intensas no mapa de bordas, mesmo quando não coincidem nos dois resultados. Este realce de informação, mesmo quando não coincidente, faz com que alguns detalhes finos apareçam no resultado final.

Assim, assume-se que os métodos individuais de entrada possuem desvantagens e defeitos característicos de sua natureza, e através deste pós-processamento aproveita-se as informações em que ambos concordam e reduz-se falhas que cada método pode gerar. Porém, as arquiteturas KSS/KoSS propostas apresentam segmentos com brechas, e os contornos não são mais fechados.

MM-Frac utiliza várias escalas, ou seja, várias janelas de diferentes tamanhos para abranger diferentes granularidades de texturas. Com isso, os montes do mapa 3D ficam mais espessos. Quando o método de crescimento de regiões germina dos vales e vai aos montes, é possível que o contorno entre regiões possa variar na espessura dos montes e, com isso, os contornos podem errar em alguns pixels. Assim, ao agrupar as informações de textura e aglutiná-las como regiões, é possível verificar que muitas vezes o contorno traçado não é tão bem localizado. Esta característica é herdada pelo KSS/KoSS, pois a estratégia de integração, enfatiza informações coincidentes de ambas as entradas, mas não altera a forma e localização dos contornos.

O controle do KSS é baseado em informação global, tal qual o controle do MM-Frac, e segue a teoria de Treisman e colegas (Treisman et al., 1977) de que o ser humano processa informação rápida, da imagem como um todo, e só depois analisa as partes das imagens de acordo com a informação global inicial.

Toda a estratégia de integração se fiou na quantificação da informação global, que pode ser interpretada como a confiança do resultado de detecção de bordas, que foi extremamente importante para o bom resultado geral do KSS/KoSS. Esta quantificação, que faz parte do “*binding problem*”, foi o diferencial que tornou possível ponderar qual resultado de cada método individual deveria ter maior importância no resultado final.

O algoritmo de integração, mesmo tendo sido implementado em Matlab, apresentou desempenho computacional adequado (o processamento de cada imagem do BSDS dura aproximadamente 0,5 segundos). Além disso, os algoritmos KSS/KoSS são fáceis de serem reproduzidos, são totalmente automáticos e não necessitam de fase de treinamento.

Foram realizados testes em duas bases de dados, a LHI e o BSDS. O LHI possui propósito

diferenciado daquele do BSDS, e seu referencial humano é bem mais resumido, mostrando contornos de objetos e não de regiões. Logo, foram realizados apenas testes qualitativos no LHI. Estes testes mostraram que mesmo utilizando imagens em diferentes resoluções e tamanhos, com grande variação da distância entre máquina fotográfica e a cena, os resultados das arquiteturas propostas são coerentes e detectam as regiões dos vários elementos que compõe a cena.

Resultados no BSDS confirmam que a integração enfatiza a localização de fronteiras entre regiões, aumenta as métricas associadas e torna o resultado final mais próximo à percepção humana que os resultados dos métodos individuais utilizados para a geração das imagens de entrada da integração.

Ainda permanecem alguns desafios que fogem do escopo desta Tese, relativos à melhoria da segmentação pelo conhecimento semântico prévio, adentrando na compreensão da percepção humana.

Capítulo 4

Conclusões Finais e Trabalhos Futuros

Nesta Tese é proposta uma nova abordagem para o problema de segmentação automática e não-supervisionada de cenas naturais coloridas. Esta abordagem pode ser visualizada como a arquitetura genérica mostrada na Figura 4.1.

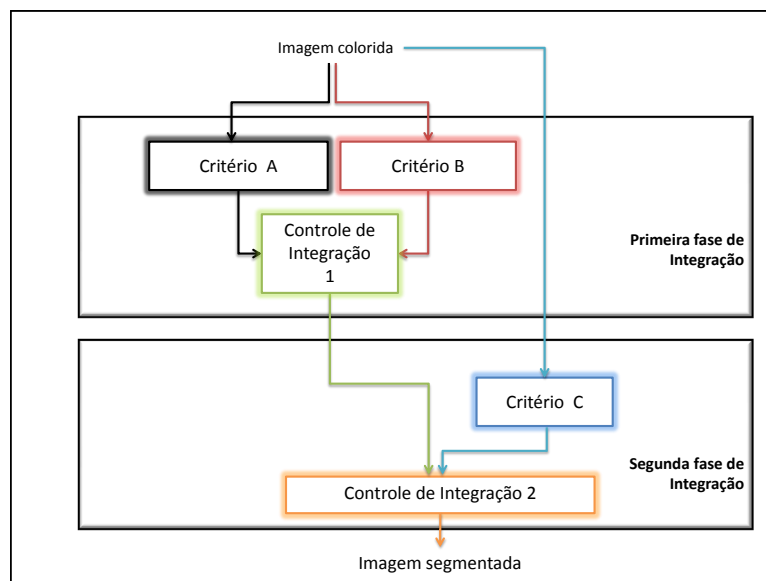


Figura 4.1: Arquitetura genérica da abordagem proposta.

A concepção desta arquitetura utilizou idéias provenientes de teorias do SHV. Assim, esta proposta aborda o uso de duas informações complementares (“*Complementary Computing*”) em estágios sequenciais (“*Hierarchical Resolution of Uncertainty*”).

Uma implementação deste “*framework*” foi testada tanto quantitativamente quanto qualitativamente. Neste sistema, a primeira fase analisa a imagem do ponto de vista de homogeneidade de cor-textura, integrando um descritor baseado em critério multifractal com um

critério baseado no discriminante linear multi-classes de Fisher. O critério de homogeneidade é usado como um mapa para o método de crescimento de regiões. O segundo estágio utiliza como informações complementares o resultado de homogeneidade (dado pela saída do primeiro estágio) e o resultado de um método de detecção de bordas. Desta forma, inspirou-se na teoria FACADE para que um caminho refletisse o processamento por superfície e o outro por borda.

Para o primeiro estágio, foram apresentadas quatro diferentes arquiteturas, bem como os resultados das segmentações em um banco de dados de imagens consolidado, discutindo-se as vantagens e desvantagens de cada uma das arquiteturas. Dois tipos diferentes de controle (“*binding*”) foram implementados, uma baseado em informação local e outro baseado em informação global. Como critério global, foi utilizado a medida de declividade da curva $1/f^\alpha$, seguindo a teoria de Treisman e colegas (Treisman et al., 1977).

Após este estágio, observou-se que o uso combinado do descritor multifractal com o *J-image* através de um controle baseado em informação global da imagem melhorou os resultados da segmentação, tornando-os mais próximos à percepção humana que os resultados do uso individual dos critérios de homogeneidade ou dos critérios combinados por controle local.

Uma abordagem comumente utilizada é criar um vetor de características, que depois é analisado por um método de agrupamento, como o *K-means*. Trabalhos anteriores já haviam utilizado descritores multifractais como um elemento num vetor de características, como no trabalho de Chaudhuri e Sarkar (1995). Nesta Tese, ao invés de se criar um vetor de características e sobrecarregar o algoritmo de agrupamento, propõe-se uma nova abordagem ao integrar diferentes medidas em um único valor, diminuindo o esforço do algoritmo de agrupamento. Também, comprovou-se o uso da medida $1/f^\alpha$ como indicador da importância da medida multifractal comparada à outra medida de homogeneidade.

A arquitetura da primeira fase de integração que apresentou o maior valor de *F-measure*, o método MM-Frac, diminuiu a supersegmentação existente em áreas de mesma textura, distinguiu melhor texturas complexas, identificou texturas de granularidade grossa e, de forma geral, localizou melhor as fronteiras entre regiões. No entanto, ainda há problemas, como a segmentação de áreas com gradação suave e a perda de contorno de informações pequenas e finas.

Para o segundo estágio, apresentou-se o KSS/KoSS, uma nova forma de integrar os resultados de dois métodos baseados em diferentes abordagens: similaridade e descontinuidade. A integração da segunda fase mantém a idéia do FACADE, de melhorar o resultado final através do uso das propriedades complementares das entradas. Esta fase também usa uma

informação global para controle da integração, que é extraída do resultado do método de detecção de bordas.

Para a escolha do método de descontinuidade, utilizou-se uma análise quantitativa, baseada nas métricas *recall*, *precision* e *F-measure*, comparada a percepção humana. Dentre os detectores clássicos com resposta não-binária: Laplace, LoG, Prewitt, Sobel e Gradiente Morfológico, o método que apresentou melhores métricas foi o gradiente morfológico. Um pré-processamento feito pelo filtro de Kuwahara melhorou ainda mais os valores das métricas.

Ao final deste estágio, é possível verificar que os problemas existentes ao final da primeira fase, a segmentação de áreas com gradação suave e a perda de contorno de informações finas, foram sanados. Ainda observa-se melhoria quanto a diminuição dos contornos falsos originados pelo método de crescimento de regiões e da diminuição do ruído originado pelo método de detecção de bordas. Ao mesmo tempo, as bordas coincidentes em ambos os resultados complementares foram enfatizadas.

Assim, os resultados finais tornaram-se mais próximos à percepção humana, comparados com os resultados dos métodos individuais utilizados como entradas da integração. Infelizmente, o KSS/KoSS apresentam segmentos com brechas, e, logo, os contornos não são mais fechados.

Comprovou-se, através desta implementação da arquitetura proposta, que é possível desenvolver fases de integração que melhoram progressivamente os resultados das detecções das fronteiras em comparação com os métodos individuais. Esta melhoria se baseia em uma comparação com a percepção média humana.

4.1 Trabalhos Futuros

A partir da estrutura genérica da solução proposta, é possível trocar cada processo da Figura 4.1 representada pelos retângulos, gerando as seguintes possibilidades de trabalhos futuros:

- estudar e testar outros critérios de homogeneidade para serem usados como “Critério A” e “Critério B”;
- testes com outros métodos detectores de bordas como “Critério C”;
- testes de outros filtros do tipo EPS, a serem usados como pré-processamento do “Critério C”;

- transformar o “Critério C” em multi-escala;
- testes com outros tipos de controle, como lógica Fuzzy ou redes neurais, no “Controle de Integração 1” e no “Controle de Integração 2”.

Pode-se ampliar o tipo de entrada do sistema, como usar imagens estéreo ou vídeo, permitindo aprofundar-se nas teorias de SHV, abrangendo características como movimento e profundidade. Outra possibilidade seria estender a arquitetura, por exemplo estudando a quantidade de estágios de integração.

Mesmo considerando as escolhas feitas para a implementação específica deste trabalho, pode-se ainda:

- avaliar outras formas de cálculo do critério multifractal;
- desenvolver variações para a escolha dos *seeds* do método de crescimento de regiões;
- avaliar outros algoritmos baseados em crescimento de região, como o *split-merge* ou *watershed*;
- avaliar outros espaços de cores;
- incluir um método de “*edge linking*” para sanar o problema das brechas nos contornos;
- avaliar o resultado quantitativo utilizando outras métricas estatísticas (Demšar, 2006; Japkowicz e Shah, 2011);
- otimizar os algoritmos para a melhoria do tempo computacional.

4.2 Trabalhos Publicados

A seguir estão listadas as publicações decorrentes desta pesquisa, em ordem cronológica:

1. Komati, K. S., Salles, E. O. T., e Sarcinelli-Filho, M. (2009). **Fractal-JSEG: JSEG using an homogeneity measurement based on local fractal descriptor**. Em Nonato, L. G. e Scharcanski, J., editors, Proceedings of the Brazilian Symposium on Computer Graphics and Image Processing, 22nd SIBGRAPI, páginas 253-260, Los Alamitos. IEEE Computer Society.

2. Komati, K. S., Salles, E. O. T., e Sarcinelli-Filho, M. (2010). **Unsupervised color image segmentation based on local fractal descriptor an *J*-images**. Em Proceedings of the IEEE ICIT 2010 International Conference on Industrial Technology, volume 1, páginas 303-308.
3. Komati, K. S., Salles, E. O. T., e Sarcinelli-Filho, M. (2010). **Unsupervised color image segmentation based on local fractal descriptor**. Em Proceedings of the 17th International Conference on Systems, Signals and Image Processing (IWSSIP 2010), volume 1, páginas 243-246, Los Alamitos. IEEE Computer Society.
4. Komati, K. S., Salles, E. O., e Sarcinelli-Filho, M. (2011). KSS: Using region and edge maps to detect image boundaries. **IEEE Computing in Science and Engineering**, 13:46-52.
5. Komati, K. S., Salles, E. O., e Sarcinelli-Filho, M. (2011). **Two-level strategy for image boundary detection. in: International conference on computer vision theory and applications**. Em Proceedings of International Conference on Computer Vision Theory and Applications (VISAPP 2011), páginas 181-186. INSTICC Press.
6. Komati, K. S., Samatelo, J. L. A., Salles, E. O. T., e Sarcinelli Filho, M. (2011). **A strategy for boundary detection combining region and edge information**. Em Lewiner, T. e Torres, R., editors, Proceedings of the 24th Brazilian Symposium on Computer Graphics and Image Processing (SIBGRAPI 2011), Los Alamitos. Conference on Graphics, Patterns and Images, 24. (SIBGRAPI), IEEE Computer Society Conference Publishing Services.
7. Komati, K., Salles, E. O. T., e Filho, M. S. (2011). **Um sistema bio-inspirado para detecção de contornos em imagens coloridas**. Em Anais do X Simpósio Brasileiro de Automação Inteligente, páginas 791-796, Universidade Federal de São João del-Rei, São João del-Rei, MG, Brasil.

Referências Bibliográficas

- Attali, D., Sanniti di Baja, G., e Thiel, E. (1997). Skeleton simplification through non significant branch removal. *Image Processing and Communications*, 3(3-4):63–72.
- Backes, A. R. (2010). *Estudos de Métodos de Análise de Complexidade em Imagens*. Tese de Doutorado, Universidade de São Paulo, campus São Carlos, Brasil.
- Barros Filho, M. e Sobreira, F. (2008). Accuracy of lacunarity algorithms in texture classification of high spatial resolution images from urban areas. Em *Proceedings of the XXI Congress of the International Society for Photogrammetry and Remote Sensing*, páginas 417–422.
- Basso, E. W. (2005). Papers e works. Website. <http://www.inf.ufrgs.br/~ewbasso/>. Acessado em: 01 set. 2011.
- Carson, C., Thomas, M., Belongie, S., Hellerstein, J. M., e Malik, J. (1999). Blobworld: a system for region-based image indexing and retrieval (long version). Relatório Técnico UCB/CSD-99-1041, EECS Department, University of California, Berkeley.
- Celebi, M. E., Iyatomi, H., Schaefer, G., e Stoecker, W. V. (2010). Lesion border detection in dermoscopy images. *Computerized medical imaging and graphics the official journal of the Computerized Medical Imaging Society*, 33(2):148–153.
- Chan, F., Lam, F., Poon, P., Zhu, H., e Chan, K. (1996). Object boundary location by region and contour deformation. *IEE Proceedings - Vision, Image, and Signal Processing*, 143(6):353–360.
- Chandran, K., Udaykumar, H., e Reinhardt, J. (2010). *Image-Based Computational Modeling of the Human Circulatory and Pulmonary Systems: Methods and Applications*. Springer.
- Chang, Y., Lee, D.-J., e Wang, Y. (2007a). Color-texture segmentation of medical images based on local contrast information. Em *Proceedings of the 2007 IEEE Symposium on Computational Intelligence in Bioinformatics and Computational Biology, (CIBCB 2007)*, páginas 488–493.

- Chang, Y. C., Archibald, J. K., Wang, Y. G., e Lee, D. J. (2007b). Texture-based color image segmentation using local contrast information. *International Journal of Information Technology and Intelligent Computing*, 2(4).
- Chaudhuri, B. B. e Sarkar, N. (1995). Texture segmentation using fractal dimension. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(1):72–77.
- Cheng, H. D., Jiang, X. H., Sun, Y., e Wang, J. (2001). Color image segmentation: advances and prospects. *Pattern Recognition*, 34(12):2259–2281.
- Chávez, P., Yarlequé, C., Piro, O., Posadas, A., Mares, V., Loayza, H., Chuquillanqui, C., Zorogastúa, P., Flexas, J., e Quiroz, R. (2010). Applying multifractal analysis to remotely sensed data for assessing PYVV infection in potato (*Solanum tuberosum* L.) crops. *Remote Sensing*, 2(5):1197–1216.
- Côco, K. F. (2007). *Modelos Fractais para a Função de Vizinhaça na Análise Topográfica de Componentes Independentes*. Tese de Doutorado, Universidade Federal do Espírito Santo, Brasil.
- Côco, K. F., Salles, E. O. T., e Sarcinelli-Filho, M. (2007). Topographic independent component analysis based on fractal theory and morphology applied to texture segmentation. *Signal Process*, 87(8):1966–1977.
- Côco, K. F., Salles, E. O. T., e Sarcinelli-Filho, M. (2009). Topographic independent component analysis based on fractal and morphology applied to texture segmentation. *Lecture Notes in Computer Science*, 5441:491–498.
- Comaniciu, D. e Meer, P. (2002). Mean shift: a robust approach toward feature space analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(5):603–619.
- Conci, A. e Nunes, E. O. (2007). Segmentação por textura e localização do contorno de regiões em imagens multibandas. *IEEE Latin America Transactions*, 5(3):184–191.
- Conci, A. e Proenca, C. (1997). A box-counting approach to color segmentation. Em *Proceedings of the 1997 International Conference on Image Processing (ICIP '97)*, volume 1 of *ICIP '97*, páginas 228–230, Washington, DC, USA. IEEE Computer Society.
- Creutzburg, R. e Ivanov, E. (1989). Fast algorithm for computing fractal dimensions of image segments. Em Cantoni, V., Creutzburg, R., Levialdi, S., e Wolf, G., editors, *Recent Issues in Pattern Analysis and Recognition*, volume 399 of *Lecture Notes in Computer Science*, páginas 42–51. Springer Berlin / Heidelberg.

- Davis, J. e Goadrich, M. (2006). The relationship between precision-recall and roc curves. Em *Proceedings of the 23rd international conference on Machine learning (ICML '06)*, páginas 233–240, New York, NY, USA. ACM.
- Demšar, J. (2006). Statistical comparisons of classifiers over multiple data sets. *Journal of Machine Learning Research*, 7:1–30.
- Deng, Y., Kenney, C., Moore, M., e Manjunath, B. S. (1999a). Peer group filtering and perceptual color image quantization. Em *Proceedings of the 1999 IEEE International Symposium on Circuits and Systems. ISCAS '99.*, volume 4, páginas 21–24.
- Deng, Y. e Manjunath, B. S. (2001). Unsupervised segmentation of color-texture regions in images and video. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(8):800–810.
- Deng, Y., Manjunath, B. S., e Shin, H. (1999b). Color image segmentation. Em *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition CVPR'99*, volume 2, páginas 446–51, Los Alamitos, CA, USA. IEEE Computer Society.
- Drummond, C. e Holte, R. C. (2004). What roc curves can't do (and cost curves can). Em *ROC Analysis in Artificial Intelligence, 1st International Workshop (ROCAI-2004)*, páginas 19–26.
- Du, G. e Yeo, T. (2002). A novel multifractal estimation method and its application to remote image segmentation. *IEEE Transactions on Geoscience and Remote Sensing*, 40(4):980–982.
- Duda, R. O. e Hart, P. E. (1973). *Pattern Classification and scene analysis*. John Wiley & Sons, Inc., New-York, USA.
- Duda, R. O., Hart, P. E., e Stork, D. G. (2001). *Pattern classification*. John Wiley and Sons, Inc, New-York, USA.
- Estrada, F. J. e Jepson, A. D. (2009). Benchmarking image segmentation algorithms. *International Journal of Computer Vision*, 85:167–181.
- Everingham, M., Van Gool, L., Williams, C. K. I., Winn, J., e Zisserman, A. (2007). The PASCAL Visual Object Classes Challenge 2007 (VOC2007) Results. <http://www.pascal-network.org/challenges/VOC/voc2007/workshop/index.html>.
- Everingham, M., Van Gool, L., Williams, C. K. I., Winn, J., e Zisserman, A. (2008). The PASCAL Visual Object Classes Challenge 2008 (VOC2008) Results. <http://www.pascal-network.org/challenges/VOC/voc2008/workshop/index.html>.

- Everingham, M., Van Gool, L., Williams, C. K. I., Winn, J., e Zisserman, A. (2009). The PASCAL Visual Object Classes Challenge 2009 (VOC2009) Results. <http://www.pascal-network.org/challenges/VOC/voc2009/workshop/index.html>.
- Everingham, M., Van Gool, L., Williams, C. K. I., Winn, J., e Zisserman, A. (2010). The PASCAL Visual Object Classes Challenge 2010 (VOC2010) Results. <http://www.pascal-network.org/challenges/VOC/voc2010/workshop/index.html>.
- Everingham, M., Van Gool, L., Williams, C. K. I., Winn, J., e Zisserman, A. (2011). The PASCAL Visual Object Classes Challenge 2011 (VOC2011) Results. <http://www.pascal-network.org/challenges/VOC/voc2011/workshop/index.html>.
- Fauzi, M. F. A. e Lewis, P. H. (2003). A fully unsupervised texture segmentation algorithm. *Proceedings of the British Machine Vision*, páginas 519–528.
- Fawcett, T. (2006). An introduction to ROC analysis. *Pattern Recognition Letters*, 27:861–874.
- Fjørtoft, R., Cabada, J. C., Lopès, A., Marthon, P., e Cubero-Castan, E. (1997). Complementary edge detection and region growing for SAR image segmentation. Em *Proceedings of Conference of the Norwegian Society for Image Processing and Pattern Recognition (NO-BIM'97)*, páginas 70–72, Tromsø, Norway.
- Frei, W. e Chung-Ching, C. (1977). Fast boundary detection: A generalization and a new algorithm. *IEEE Transactions on Computers*, 26:988–998.
- Frisch, U. (1980). Fully developed turbulence and intermittency. *Annals of the New York Academy of Sciences*, 357:359–367.
- Fua, P. e Hanson, A. J. (1987). Using generic geometric models for intelligent shape extraction. Em *Proceedings of the sixth National conference on Artificial intelligence*, volume 2 of AAAI'87, páginas 706–711. AAAI Press.
- Gagalowicz, A. e Monga, O. (1986). A new approach for image segmentation. Em *Proceedings of the International Conference on Pattern Recognition*, páginas 1–6. IEEE Computer Society.
- Gersho, A. e Gray, R. M. (1992). *Vector quantization and signal compression*. Kluwer, Boston.
- Gonzalez, R. C. e Woods, R. E. (2001). *Digital Image Processing*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 2nd edição.

- Gordon, M. e Kochen, M. (1989). Recall-precision trade-off: A derivation. *Journal of the American Society for Information Science*, 40(3):145–151.
- Grossberg, S. (2000a). The complementary brain: unifying brain dynamics and modularity. *Trends in Cognitive Sciences*, 4(6):233–246.
- Grossberg, S. (2000b). Linking mind to brain: The mathematics of biological intelligence. *Notices of the American Mathematical Society*, 47(11):1361–1372.
- Grossberg, S. (2006). The art of seeing and painting. Relatório Técnico CAS/CNS-2006-011, Department of Cognitive and Neural Systems, Boston University.
- Grossberg, S. e Mingolla, E. (1985). Neural dynamics of perceptual grouping: textures, boundaries, and emergent segmentations. *Percept Psychophys*, 38(2):141–171.
- Gunther, N. J. (2005). Millennium performance problem 1: Performance visualization. Web-site. http://www.cmg.org/measureit/issues/mit22/m_22_1.html. Acessado em: 01 set. 2011.
- Haindl, M. e Mikes, S. (2008). Texture segmentation benchmark. Em *Proceedings of the 19th International Conference on Pattern Recognition (ICPR 2008)*, páginas 1–4.
- Holcombe, A. O. (2009). The binding problem. Em Goldstein, B., editor, *The Sage Encyclopedia of Perception*. SAGE Publications, Inc.
- Houqiang, L., Zhengkai, L., e Feng, L. (2001). Texture image segmentation method based on fractal theory and kohonen neural network. *Computer Engineering and Applications*, 37:44–46.
- Hsu, T.-I., Kuo, J. L., e Wilson, R. (2000). A multiresolution texture gradient method for unsupervised segmentation. *Pattern Recognition*, 33(11):1819–1833.
- Ivanov, P. C., Amaral, L. A. N., Goldberger, A. L., Havlin, S., Rosenblum, M. G., Struzik, Z. R., e Stanley, H. E. (1999). Multifractality in human heartbeat dynamics. *Nature*, 399:461–465.
- Jain, A. K. (1989). *Fundamentals of digital image processing*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA.
- Jain, A. K., Duin, R. P. W., e Mao, J. (2000). Statistical pattern recognition: A review. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(1):4–37.
- Japkowicz, N. e Shah, M. (2011). *Evaluating Learning Algorithms: A Classification Perspective*. Cambridge University Press, New York, NY, USA.

- Kandel, E. R., Schwartz, J. H., e Jessell, T. M. (2000). *Principles of neural science*. McGraw-Hill, New York.
- Kasparis, T., Charalampidis, D., Georgiopoulos, M., e Rolland, J. P. (2001). Segmentation of textured images based on fractals and image filtering. *Pattern Recognition*, 34(10):1963–1973.
- Komati, K., Salles, E. O. T., e Filho, M. S. (2011a). Um sistema bio-inspirado para detecção de contornos em imagens coloridas. Em *Anais do X Simpósio Brasileiro de Automação Inteligente*, páginas 791–796, Universidade Federal de São João del-Rei, São João del-Rei, MG, Brasil.
- Komati, K. S., Ferreira, A., e Souza, D. (2003). Vergence control in a binocular vision system using weightless neural networks. *Applied Bionics and Biomechanics*, 1(1):21–31.
- Komati, K. S., Salles, E. O., e Sarcinelli-Filho, M. (2011b). KSS: Using region and edge maps to detect image boundaries. *Computing in Science and Engineering*, 13:46–52.
- Komati, K. S., Salles, E. O., e Sarcinelli-Filho, M. (2011c). Two-level strategy for image boundary detection. in: International conference on computer vision theory and applications. Em *Proceedings of International Conference on Computer Vision Theory and Applications (VISAPP 2011)*, páginas 181–186. INSTICC Press.
- Komati, K. S., Salles, E. O. T., e Sarcinelli-Filho, M. (2009). Fractal-JSEG: JSEG using an homogeneity measurement based on local fractal descriptor. Em Nonato, L. G. e Scharcanski, J., editors, *Proceedings of the Brazilian Symposium on Computer Graphics and Image Processing, 22nd SIBGRAPI*, páginas 253–260, Los Alamitos. IEEE Computer Society.
- Komati, K. S., Salles, E. O. T., e Sarcinelli-Filho, M. (2010a). Unsupervised color image segmentation based on local fractal descriptor. Em *Proceedings of the 17th International Conference on Systems, Signals and Image Processing (IWSSIP 2010)*, volume 1, páginas 243–246, Los Alamitos. IEEE Computer Society.
- Komati, K. S., Salles, E. O. T., e Sarcinelli-Filho, M. (2010b). Unsupervised color image segmentation based on local fractal descriptor an J-images. Em *Proceedings of the IEEE ICIT 2010 International Conference on Industrial Technology*, volume 1, páginas 303–308.
- Komati, K. S., Samatelo, J. L. A., Salles, E. O. T., e Sarcinelli Filho, M. (2011d). A strategy for boundary detection combining region and edge information. Em Lewiner, T. e Torres, R., editors, *Proceedings of the 24th Brazilian Symposium on Computer Graphics and*

- Image Processing (SIBGRAPI 2011)*, Los Alamitos. Conference on Graphics, Patterns and Images, 24. (SIBGRAPI), IEEE Computer Society Conference Publishing Services.
- Kuwahara, M., Hachimura, K., Eiho, S., e Kinoshita, M. (1976). *Digital processing of biomedical images*, páginas 187–203. Plenum Press.
- Lam, L. (1998). *Nonlinear Physics for Beginners: Fractals, Chaos, Solitons, Pattern Formation, Cellular Automata and Complex Systems*. World Scientific.
- Lashermes, B., Roux, S. G., Abry, P., e Jaffard, S. (2008). Comprehensive multifractal analysis of turbulent velocity using the wavelet leaders. *The European Physical Journal B - Condensed Matter and Complex Systems*, 61(2):201–215.
- Liu, Y., Zhang, D., Lu, G., e Ma, W.-Y. (2007). A survey of content-based image retrieval with high-level semantics. *Pattern Recognition*, 40:262–282.
- Lopes, R. e Betrouni, N. (2009). Fractal and multifractal analysis: a review. *Medical Image Analysis*, 13(4):634–649.
- Maeda, J., Ishikawa, C., Novianto, S., Tadehara, N., e Suzuki, Y. (2000). Rough and accurate segmentation of natural color images using fuzzy region-growing algorithm. Em *Proceedings of the 15th International Conference on Pattern Recognition (ICPR'00)*, volume 3, páginas 3642–3645.
- Mandelbrot, B. B. (1983). *The Fractal Geometry of Nature*. W. H. Freedman and Co., New York.
- Martin, D. (2002). *An Empirical Approach to Grouping and Segmentation*. Tese de Doutorado, University of California, Berkeley.
- Martin, D., Fowlkes, C., Tal, D., e Malik, J. (2001). A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. Em *Proceedings of the 8th IEEE International Conference on Computer Vision*, volume 2, páginas 416–423.
- Martin, D. R., Fowlkes, C. C., e Malik, J. (2004). Learning to detect natural image boundaries using local brightness, color, and texture cues. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(5):530–549.
- Mishkin, M. e Ungerleider, L. G. (1982). Contribution of striate inputs to the visuospatial functions of parieto-preoccipital cortex in monkeys. *Behavioural Brain Research*, 6(1):57–77.
- Mitchell, T. M. (1997). *Machine Learning*. McGraw-Hill International Editions.

- Muñoz, X., Freixenet, J., Cufí, X., e Martí, J. (2003). Strategies for image segmentation combining region and boundary information. *Pattern Recognition Letters*, 24(1-3):375–392.
- Nunes, É. d. O. e Conci, A. (2003). Texture segmentation considering multi band, multi resolution and affine invariant roughness. Em *Proceedings of the 16th Brazilian Symposium on Computer Graphics and Image Processing (SIBGRAPI 2003)*, páginas 254–261. IEEE Computer Society.
- Olshausen, B. A. e Field, D. J. (2005). How close are we to understanding v1? *Neural Computation*, 17:1665–1699.
- Otsu, N. (1979). A threshold selection method from gray-level histograms. *IEEE Transactions on Systems, Man and Cybernetics*, 9(1):62–66.
- Ozden, M. e Polat, E. (2007). A color image segmentation approach for content-based image retrieval. *Pattern Recognition*, 40(4):1318–1325.
- Palmer, S. E. (1999). *Vision science : photons to phenomenology*. MIT Press, Cambridge, Massachusets.
- Pavlidis, T. e Liow, Y.-T. (1990). Integrating region growing and edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12:225–233.
- Peleg, S., Naor, J., Hartley, R., e Avnir, D. (1984). Multiple resolution texture analysis and classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6(4):518–523.
- Pentland, A. P. (1984). Fractal-based description of natural scenes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6(6):661–674.
- Pesquet-Popescu, B. e Lévy Véhel, J. (2002). Stochastic Fractal Models for Image Processing. *IEEE Signal Processing Magazine*, 19(5):48–62.
- Plamen, C. I., Amaral, L. A. N., Goldberger, A. L., Havlin, S., Rosenblum, M. G., Stanley, H. E., e Struzik, Z. R. (2001). From 1/f noise to multifractal cascades in heartbeat dynamics. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 11(3):641–652.
- Pres, W. H. (2008). Cheat sheet on accuracy, precision, recall, tpr, fpr, specificity, sensitivity, roc, and all that stuff! Website. www.nr.com/whp/ConfusionMatrixDefns.pdf. Acessado em: 01 set. 2011.

- Rao, S., Mobahi, H., Yang, A. Y., Sastry, S., e Ma, Y. (2009). Natural image segmentation with adaptive texture and boundary encoding. Em Zha, H., ichiro Taniguchi, R., e Maybank, S. J., editors, *Proceedings of 9th Asian Conference on Computer Vision, ACCV'09*, volume 5994 of *Lecture Notes in Computer Science*, páginas 135–146. Springer.
- Revathy, K. (2007). Feature recognition and classification using spectral methods. Em Zharkova, V. e Jain, L., editors, *Artificial Intelligence in Recognition and Classification of Astrophysical and Medical Images*, volume 46 of *Studies in Computational Intelligence*, páginas 339–374. Springer Berlin / Heidelberg. 10.1007/978-3-540-47518-7_5.
- Revol-Muller, C., Peyrin, F., Odet, C., e Carillon, Y. (2000). Automated 3D region growing algorithm governed by an evaluation function. Em *Proceedings of the 2000 International Conference on Image Processing (ICIP 2000)*.
- Rotem, O., Greenspan, H., e Goldberger, J. (2007). Combining region and edge cues for image segmentation in a probabilistic gaussian mixture framework. Em *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2007)*.
- Russell, B., Torralba, A., Murphy, K., e Freeman, W. (2008). LabelMe: A Database and Web-Based Tool for Image Annotation. *International Journal of Computer Vision*, 77(1):157–173.
- Saeed, U. e Dugelay, J.-L. (2010). Combining edge detection and region segmentation for lip contour extraction. Em *Proceedings of the 6th international conference on Articulated motion and deformable objects, AMDO'10*, páginas 11–20, Berlin, Heidelberg. Springer-Verlag.
- Sarkar, N. e Chaudhuri, B. B. (1994). An efficient differential box-counting approach to compute fractal dimension of image. *IEEE Transactions on Systems Man and Cybernetics*, 24(1):115–120.
- Sarkar, N. e Chaudhuri, B. B. (1995). Multifractal and generalized dimensions of gray-tone digital images. *Signal Processing*, 42(2):181–190.
- Schiller, P. H., True, S. D., e Conway, J. L. (1980). Deficits in eye movements following frontal eye-field and superior colliculus ablations. *Journal of Neurophysiology*, 44(6):1175–1189.
- Sclaroff, S. e Liu, L. (2001). Deformable shape detection and description via model-based region grouping. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23:475–489.

- Shotton, J., Winn, J., Rother, C., e Criminisi, A. (2006). Textonboost: Joint appearance, shape and context modeling for multi-class object recognition and segmentation. Em Leonardis, A., Bischof, H., e Pinz, A., editors, *Computer Vision (ECCV 2006)*, volume 3951 of *Lecture Notes in Computer Science*, páginas 1–15. Springer Berlin / Heidelberg. 10.1007/11744023_1.
- Spann, M. e Wilson, R. (1985). A quad-tree approach to image segmentation which combines statistical and spatial information. *Pattern Recognition*, 18(3-4):257–269.
- Staal, J., Abramoff, M., Niemeijer, M., Viergever, M., e van Ginneken, B. (2004). Ridge based vessel segmentation in color images of the retina. *IEEE Transactions on Medical Imaging*, 23(4):501–509.
- Stanley, H. E. e Meakin, P. (1988). Multifractal phenomena in physics and chemistry. *Nature*, 335(6189):405–409.
- Teng, H. T., Ewe, H.-T., e Tan, S. L. (2010). Multifractal dimension and its geometrical terrain properties for classification of multi-band multi-polarized sar image. *Progress In Electromagnetics Research*, 104:221–237.
- Theiler, J. (1990). Estimating fractal dimension. *Journal of the Optical Society of America*, 7(6):1055–1073.
- Torralba, A. e Oliva, A. (2003). Statistics of natural image categories. *Network: Computation in Neural Systems*, 14(3):391–412.
- Treisman, A. M., Sykes, M., e Gelade, G. (1977). Selective attention and stimulus integration. Em Dornic, S., editor, *Attention and Performance VI*, chapter 17. Lawrence Erlbaum.
- Tu, Z. e Zhu, S.-C. (2002). Image segmentation by data-driven markov chain monte carlo. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24:657–673.
- Turner, M. J., Andrews, P. R., e Blackledge, J. M. (1998). *Fractal Geometry in Digital Imaging*. Academic Press, Inc., Orlando, FL, USA, 1st edição.
- Véhel, J. L. e Mignot, P. (1994). Multifractal Segmentation of Images. *Fractals*, 2(3):371–378.
- Ventura, V. A. (2009). Representação de imagens através de grafos utilizando o algoritmo split-and-merge combinado com descritores de cor e textura. Dissertação de Mestrado, Universidade Federal do Espírito Santo, Brasil.
- Voss, R. F. (1986). Random fractals: characterization and measurement. *Scaling Phenomena in Disordered Systems*, R. Pynn and Skjeltorp Eds.

- Wang, Y.-G., Yang, J., e Chang, Y.-C. (2006). Color-texture image segmentation by integrating directional operators into JSEG method. *Pattern Recognition Letters*, 27(16):1983–1990.
- Wilson, R. e Spann, M. (1988). Finite prolate spheroidal sequences and their applications II: Image feature description and segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 10(2):193–203.
- Wilson, R. e Spann, M. (1990). A new approach to clustering. *Pattern Recognition*, 23(12):1413–1425.
- Wolf, C. e Jolion, J.-M. (2006). Object count/area graphs for the evaluation of object detection and segmentation algorithms. *International Journal on Document Analysis and Recognition*, 8:280–296.
- Xia, Y., Feng, D. D., e Zhao, R. (2006). Morphology-based multifractal estimation for texture segmentation. *IEEE Transactions on Image Processing*, 15(3):614–623.
- Yao, B., Yang, X., e Zhu, S.-C. (2007). Introduction to a large-scale general purpose ground truth database: Methodology, annotation tool and benchmarks. Em Yuille, A., Zhu, S.-C., Cremers, D., e Wang, Y., editors, *Energy Minimization Methods in Computer Vision and Pattern Recognition*, volume 4679 of *Lecture Notes in Computer Science*, páginas 169–183. Springer Berlin / Heidelberg. 10.1007/978-3-540-74198-5_14.
- Zhang, H., Fritts, J. E., e Goldman, S. A. (2008). Image segmentation evaluation: A survey of unsupervised methods. *Computer Vision and Image Understanding*, 110:260–280.
- Zharkova, V. V. e Jain, L. C., editors (2007). *Artificial Intelligence in Recognition and Classification of Astrophysical and Medical Images*, volume 46 of *Studies in Computational Intelligence*. Springer.

Apêndice A

JSEG

O método JSEG (Deng e Manjunath, 2001; Deng et al., 1999b) tem como objetivo segmentar imagens e vídeos de cenas naturais em regiões homogêneas de cor-textura. Neste apêndice se descreve detalhadamente como o método funciona para imagens, mas não se descreve a parte do método que trata vídeo.

Esta abordagem não tenta estimar parâmetros de um modelo de textura específico, mas sim, testar a homogeneidade de um dado padrão de acordo com a sua cor e textura. Para identificar esta homogeneidade, pressupõe-se que:

- cada imagem contém um conjunto finito de regiões homogêneas baseadas nas suas cores e texturas;
- as informações de cores em cada região da imagem podem ser representadas por um conjunto de poucas cores quantizadas, e
- as cores entre duas regiões vizinhas são distinguíveis.

A idéia básica do método JSEG é separar o processo de segmentação em dois estágios: quantização de cores e segmentação espacial. No primeiro estágio, as cores na imagem são quantizadas em algumas classes representativas de cores que podem ser usadas para diferenciar regiões em uma imagem. Esta quantização é feita no espaço de cores, sem levar em consideração a distribuição espacial das cores na imagem. Depois, cada pixel da imagem é substituído pelo rótulo de sua classe de cores correspondente, formando um mapa-de-classes da imagem. No segundo estágio, a segmentação espacial é processada diretamente no mapa-de-classes, sem considerar a similaridade entre as cores dos pixels. Uma visão geral de tal sistema pode ser vista na Figura A.1a.

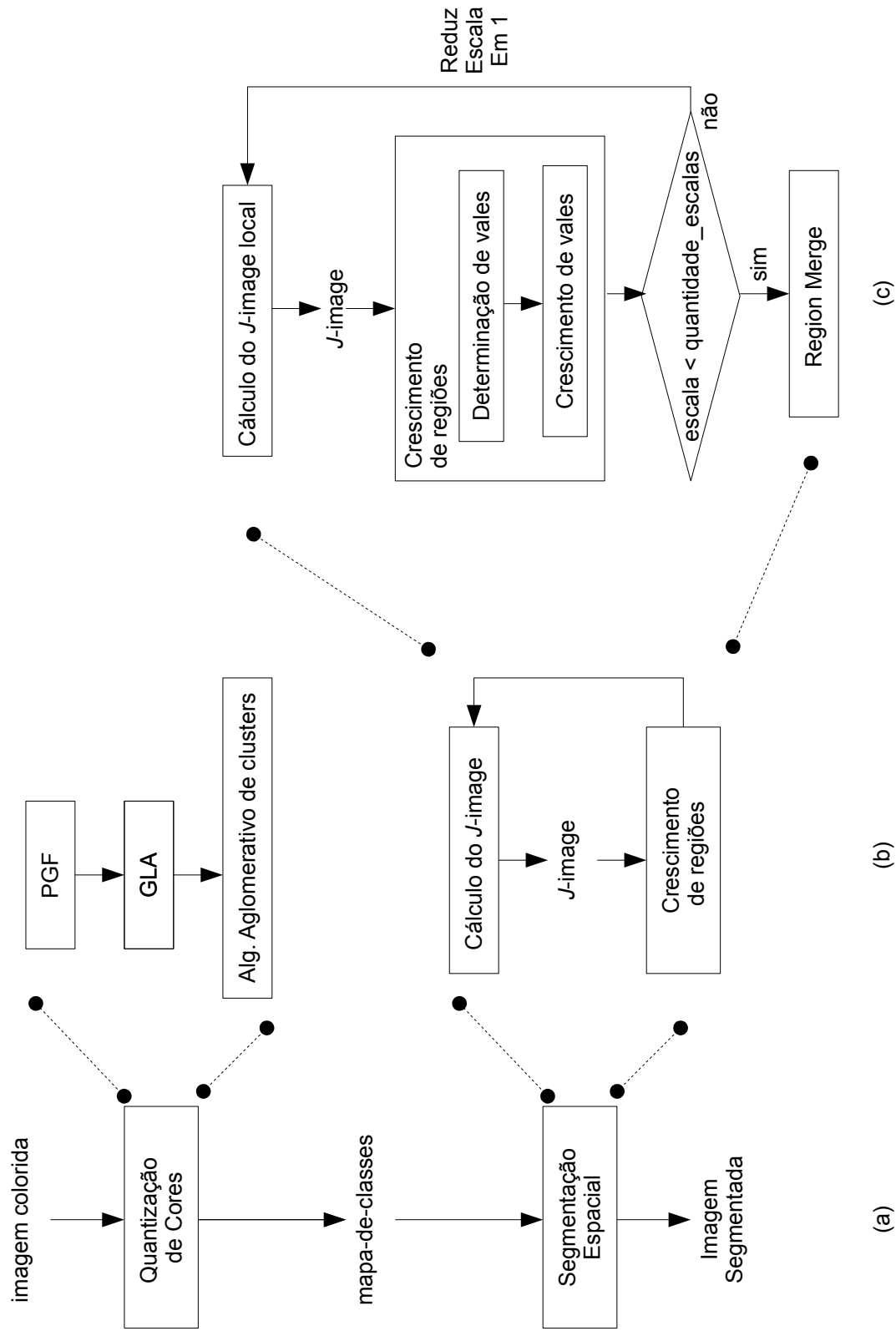


Figura A.1: Arquitetura em dois estágios do JSEG. (a) visão básica (b) visão mais detalhada de (a) (c) apresentando algoritmo de segmentação espacial com o controle de quantidade de escalas.

Do ponto de vista dos autores, a separação em dois estágios tem um benefício evidente, que é simplificar a tarefa, primeiro analisando apenas a similaridade de cores e depois apenas as suas distribuições, e não as duas características ao mesmo tempo. A dissociação dos conceitos de similaridade de cores com a distribuição espacial permite o desenvolvimento de algoritmos mais tratáveis para cada um dos dois estágios. Cada um destes estágios será descrito a seguir, nas seções A.1 e A.2.

A.1 Quantização de Cores

O objetivo desta etapa é diminuir a quantidade de cores sem degradar significativamente a qualidade da percepção visual da imagem. Uma imagem colorida de 24 bits apresenta tipicamente milhares de cores, que dificultam a manipulação. Logo, a idéia é extrair apenas algumas poucas cores representativas que possam diferenciar as regiões vizinhas de uma imagem. Os resultados dos testes do trabalho de Deng e Manjunath apresentaram quantidade final de cores, tipicamente, entre 10 a 20 cores, para imagens de cenas naturais.

Antes do processo de quantização propriamente dito, JSEG faz um pré-processamento usando o algoritmo “Peer Group Filtering” (PGF) (Deng et al., 1999a). PGF é um algoritmo não-linear para suavização da imagem e retirada do ruído impulsivo. O algoritmo se caracteriza por ser mais sensível à mudanças em áreas mais lisas do que em áreas texturizadas. Assim, as cores podem ser quantizadas de forma mais grosseira em áreas com texturas.

Os seguintes passos são executados sequencialmente, conforme pode ser visto na parte superior da Figura A.1b:

1. O algoritmo PGF é aplicado à imagem.
2. Como resultado do PGF, obtém-se valores indicando a suavidade de áreas locais. Estes pesos são associados a cada pixel, tal que são menores em áreas texturizadas do que em superfícies lisas.
3. Uma versão modificada do “General Lloyd algorithm” (GLA) (Gersho e Gray, 1992) é usado para quantizar as cores. O espaço de cores CIE LUV foi o escolhido para o processo de quantização.
4. Depois do GLA, uma grande quantidade de pixels com aproximadamente a mesma cor estarão em *clusters* diferentes. Logo, o algoritmo aglomerativo de *clusters* (Duda e Hart, 1973) é executado para juntar clusters cujos centróides satisfaçam um limiar pré-definido.

Depois da quantização, dividem-se as cores quantizadas em classes. Cada classe de cor é formada pelo conjunto de pixels que foram quantizados para a mesma cor. Cada pixel da imagem tem seu valor substituído pela informação da classe, formando o mapa-de-classes da imagem. Este mapa pode ser visto como um tipo especial de composição de texturas. Exemplos de mapas-de-classes são mostrados na Figura A.2, onde os rótulos das classes são representados pelos símbolos '+', '*' e 'o'. Na prática, os rótulos são números inteiros.

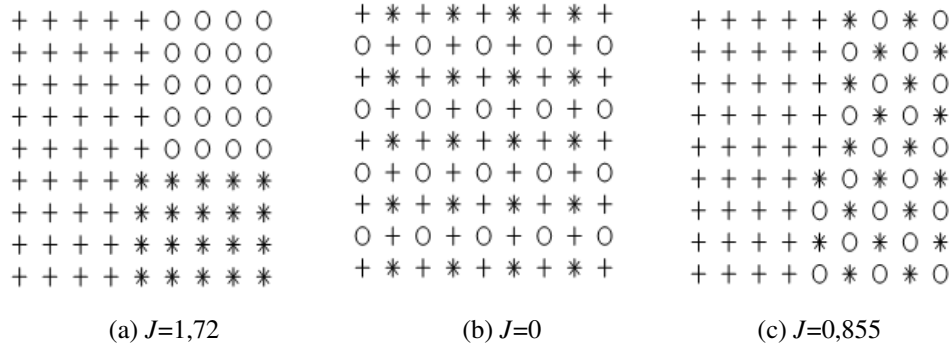


Figura A.2: Exemplos de três mapas-de-classes. Fonte: Deng e Manjunath (2001)

A.2 Segmentação Espacial

O mapa-de-classes é a imagem de entrada da segmentação espacial. A segmentação espacial também pode ser vista como uma sequência de dois passos: o cálculo do J -image e o processo de crescimento de regiões, conforme pode ser visto no diagrama de blocos da parte inferior da Figura A.1b.

A.2.1 J -image

Na segmentação, utiliza-se a medida J , que é o critério para medir a distribuição das classes de cores, definida da seguinte forma: seja Z o conjunto de todos os N pontos no mapa de classes. Define-se um ponto $z = (x, y)$, $z \in Z$, sendo m o ponto médio de todos os pontos da imagem, ou seja,

$$m = \frac{1}{N} \sum_{z \in Z} z. \quad (\text{A.1})$$

Suponha que Z é classificado em C classes, $Z_i, i = 1, \dots, C$. Faça m_i ser o ponto médio de todos os N_i elementos da classe Z_i :

$$m_i = \frac{1}{N_i} \sum_{z \in Z_i} z. \quad (\text{A.2})$$

A variância total dos pontos em Z , denominada S_T , é dada por:

$$S_T = \sum_{z \in Z} \|z - m\|^2, \quad (\text{A.3})$$

e S_W , a variância total dos pontos pertencentes à mesma classe, é definida como:

$$S_W = \sum_{i=1}^C S_i = \sum_{i=1}^C \sum_{z \in Z_i} \|z - m_i\|^2. \quad (\text{A.4})$$

Então, o valor J é definido como:

$$J = (S_T - S_W)/S_W = S_B/S_W. \quad (\text{A.5})$$

O índice W indica o termo em inglês “*within*” e B , “*between*”. Essencialmente, a equação (A.5) mede as distâncias entre as diferentes classes, S_B , dividido pelas distâncias entre os membros de cada classe S_W , uma idéia similar ao discriminante linear multi-classes de Fisher (Duda et al., 2001). Um alto valor de J indica que as classes estão mais separadas umas das outras, e que os membros de cada classe estão mais próximos uns dos outros. Por outro lado, se todas as classes estão uniformemente distribuídas sobre toda a imagem, o valor de J tende a ser pequeno. Por exemplo, para a Figura A.2a, o valor de J é 1,72, para a Figura A.2b é 0 e para a Figura A.2c é 0,855. Note-se que todos os três mapas possuem a mesma quantidade de pontos em cada uma das três classes.

Considere o mapa-de-classes 1 da Figura A.2. Uma “boa” segmentação para este caso poderia ser uma divisão em 3 regiões, cada uma contendo um rótulo único. Como o mapa-de-classes 2 é uniforme, nenhuma segmentação é necessária. Para o mapa-de-classes 3, uma “boa” segmentação seriam duas regiões, uma contendo a região com a classe ‘+’ e outra contendo as classes ‘*’ e ‘o’. A Figura A.3 mostra a segmentação manual descrita para os mapas-de-classes 1 e 3. Recalculando o valor J sobre cada região segmentada ao invés de todo o mapa-de-classe, definimos a média ^{-}J :

$$^{-}J = \frac{1}{N} \sum_k M_k J_k, \quad (\text{A.6})$$

onde J_k é o valor de J calculado sobre uma região k , M_k é o número de pontos na região k , N é o número total de pontos no mapa-de-classes, e a soma é sobre todas as regiões no mapa-de-classes.

Para um número fixo de regiões, uma “melhor” segmentação tende a ter um menor valor de \bar{J} . Se a segmentação é boa, cada região contém poucas classes distribuídas uniformemente resultando em um baixo valor de J para a região, daí, no geral \bar{J} também é menor. O menor e melhor valor possível seria 0 (zero). Os valores de \bar{J} e J_k para os mapas-de-classe 1 e 3 são mostrados na legenda da Figura A.3.

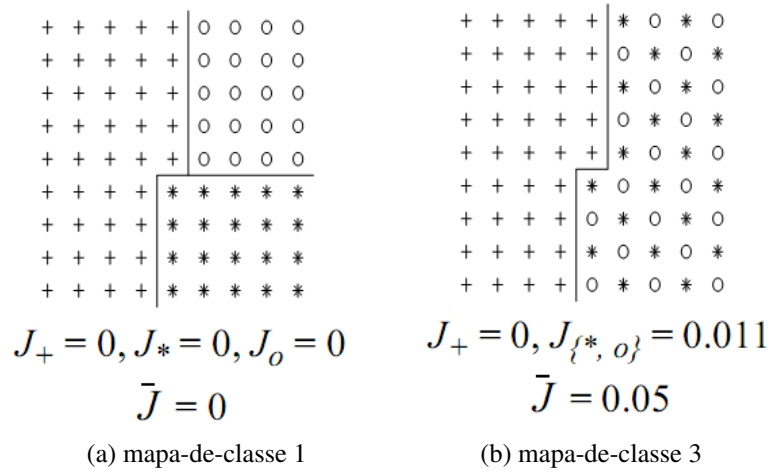


Figura A.3: “Melhor” segmentação dos mapas-de-classe 1 e 3. Fonte: Deng e Manjunath (2001)

A minimização global para toda a imagem não é prática, dado que existem milhões de formas possíveis para segmentar uma imagem. Contudo, observa-se que J aplicado a uma área local do mapa-de-classes também é um bom indicador para determinar se a área está no interior da região ou se está próxima à uma borda. Logo, pode-se criar uma imagem, J -image, cujos pixels correspondem aos valores J calculados sobre uma janela local centralizada em cada pixel, denominado por valores J -local. O tamanho da janela local determina o tamanho das regiões que podem ser detectadas. A J -image de cada escala é calculada alterando-se o tamanho da janela local.

Na J -image, quanto maior o valor de J -local maior é a probabilidade do pixel estar próximo a uma região de contorno. A J -image é como um mapa 3-D contendo vales e montanhas, que representam o centro e as fronteiras das regiões, respectivamente.

A.2.2 Crescimento de Regiões

As características da J -image permitem utilizar um método de crescimento de regiões para a segmentação da imagem. O processo de crescimento de regiões consiste em determinar as sementes e como é o crescimento em torno destas sementes. Considera-se a imagem original como a região inicial.

A Tabela A.1 mostra os tamanhos das janelas locais em cada escala, bem como é feita a sub-amostragem (*downsampling*) de cada janela para acelerar a performance computacional. Por exemplo, para a escala 2, o tamanho da janela local é 17×17 , onde a cada dois pixels aproveita-se um, tanto na direção x quanto na direção y , diminuindo-se a quantidade de informações a serem consideradas em cada janela. O algoritmo começa com uma janela local inicial grande, e repete a mesma transformação com a escala seguinte (uma janela menor), até que a escala mínima especificada seja atingida, conforme mostrado na Figura A.1c.

Tabela A.1: Tamanho das janelas para cada escala.

escala	janela (pixels)	amostragem (1/pixels)	tamanho da região (pixels)	tamanho mínimo da semente (pixels)
1	9×9	$1/(1 \times 1)$	64×64	32
2	17×17	$1/(2 \times 2)$	128×128	128
3	33×33	$1/(4 \times 4)$	256×256	512
4	65×65	$1/(8 \times 8)$	512×512	2048

Determinação das Sementes

As sementes que darão início ao processo de crescimento de regiões correspondem às áreas de mínimo dos valores J . As seguinte heurística foi seguida:

1. Calcula-se a média e o desvio padrão dos valores de J numa determinada região, μ_J e σ_J respectivamente.
2. Calcula-se um limiar T_J dado por:

$$T_J = \mu_J + a\sigma_J. \quad (\text{A.7})$$

onde a é escolhido a partir de um conjunto pré-definido de valores $[-0,6; -0,4; -0,2; 0; 0,2; 0,4]$, como aquele que irá resultar na maior quantidade de sementes. Pixels com

valores J menores que T_J são considerados como candidatos a sementes. Conectando-se a quantidade de sementes baseados em sua 4-conectividade, obtém-se as áreas sementes candidatas.

3. Se uma área semente candidata tem um valor maior que o tamanho mínimo listado na última coluna na Tabela A.1 em sua escala correspondente, então a área é marcada como semente.

O Processo de Crescimento

O processo de crescimento de regiões é feito em várias escalas. A cada escala, o tamanho da janela local diminui, oferecendo maior precisão na localização de bordas. As regiões crescem em torno das sementes, seguindo os passos:

1. Remova os “buracos” dentro das áreas semente.
2. Calcule a média dos valores J que ainda não fazem parte de alguma região e conecte os pixels abaixo desta média para formar áreas de crescimento. Se esta área de crescimento é adjacente a uma e apenas uma semente, então a área de crescimento é associada à semente em questão.
3. Pixels não classificados em alguma região são armazenados em um “buffer”. Entra-se em um laço onde, a cada vez, o pixel de menor J é associado a uma semente adjacente, até que todos os pixels sejam classificados.

Na menor escala, o crescimento de regiões é feito pixel por pixel, e não por áreas de crescimento, como no passo 2.

Region Merge

Finalmente, para diminuir o problema de supersegmentação, as regiões são mescladas (*region merge*), com base em semelhança de cores. Após a junção de regiões, a segmentação final é obtida.

A informação sobre as cores de uma região é caracterizada pelo seu histograma de cores. Cada cor resultante da quantização representa uma barra vertical do histograma. A partir do histograma de cores de cada região, calcula-se a distância Euclidiana entre duas regiões vizinhas. O par de regiões com a menor distância é mesclado e, com isso, um novo histograma é calculado para essa nova região, e recalcula-se todas as distâncias. O processo continua até que o limiar máximo seja alcançado.

Apêndice B

Precision × *Recall* e ROC

A aprendizagem de máquina (*Machine Learning*) é um sub-campo da inteligência artificial dedicado ao desenvolvimento de algoritmos e técnicas que permitam ao computador aprender, isto é, que permitam ao computador aperfeiçoar seu desempenho em alguma tarefa (Mitchell, 1997). Enquanto que na Inteligência Artificial existem dois tipos de raciocínio: indutivo, que extrai regras e padrões de grandes conjuntos de dados, e dedutivo, que é constituído de axiomas e regras de inferência que podem ser usadas para derivar os teoremas do sistema. A aprendizagem de máquina só se preocupa com o indutivo. Aplicações práticas incluem o processamento de linguagem natural, motores de busca (*search engine*), diagnósticos médicos, bioinformática, reconhecimento de fala, reconhecimento de escrita, visão computacional e locomoção de robôs.

Uma tarefa importante de aprendizado de máquina é a classificação, que também é referida como reconhecimento de padrões, em que as máquinas “aprendem” a reconhecer automaticamente padrões complexos, e assim fazer previsões quanto à qual classe pertencem novos exemplares. Alguns classificadores geram como saída valores contínuos (por exemplo, um classificador que gera a probabilidade de uma instância pertencer a uma dada classe), para os quais diferentes limiares podem ser aplicados para gerar diferentes conjuntos de saída (em um classificador contínuo pode-se aplicar um limiar para binarizá-lo, isto é, gerar duas classes). Outros, geram resultados discretos indicando somente a classe.

Para avaliar o resultado de um classificador binário, é comum usar gráficos de métricas derivados da matriz de confusão. A matriz de confusão oferece uma medida efetiva do modelo de classificação, ao mostrar o número de classificações corretas versus as classificações preditas para cada classe. Considera-se que uma instância pode assumir valores p (‘positivo’) e n (‘negativo’). Assim, há quatro possibilidades:

1. se a instância é ‘positivo’ e é classificada como ‘positivo’, conta-se como ‘verdadeiro

- positivo' (TP=*true positive*);
2. se a instância é 'positivo' e é classificada como 'negativo', conta-se como 'falso negativo' (FN=*false negative*);
 3. se a instância é 'negativo' e é classificada como 'negativo', conta-se como 'verdadeiro negativo' (TN=*true negative*) e
 4. se a instância é 'negativo' e é classificada como 'positivo', conta-se como 'falso positivo' (FP=*false positive*).

Portanto, tendo um classificador e um conjunto de instâncias pode-se construir uma matriz de confusão de 2 por 2, conforme a Tabela B.1.

Tabela B.1: Matriz de confusão para um classificador binário.

Classe	classe verdadeira (p)	classe verdadeira (n)
resposta do classificador (p)	TP	FP
resposta do classificador (n)	FN	TN
Totais das colunas	P (= TP + FN)	N (= FP + TN)

O número de acertos, para cada classe, se localiza na diagonal principal da matriz (TP e TN). Os demais elementos representam erros na classificação, FP e FN. A matriz de confusão de um classificador ideal possui FP e FN iguais a zero uma vez que ele não comete erros. A partir desta matriz pode-se derivar as seguintes métricas:

$$recall = TPR = \frac{TP}{TP + FN} = \frac{TP}{P}, \quad (B.1)$$

$$FPR = \frac{FP}{FP + TN} = \frac{FP}{N}, \quad (B.2)$$

$$precision = \frac{TP}{TP + FP}, \quad (B.3)$$

$$F\text{-measure} = \frac{2 \times recall \times precision}{recall + precision}, \quad (B.4)$$

onde TPR (TP rate) ou *recall* é a taxa de acerto do classificador e FPR (FP rate) é a taxa de alarme falso, que serão os eixos y e x , respectivamente, do espaço ROC (*Receiver operating*

characteristic) (Fawcett, 2006). A métrica *precision* é a taxa de acerto do algoritmo com relação ao total que o próprio algoritmo retornou como p . A métrica *F-measure* é a média harmônica de *precision* e *recall*.

Veja que o gráfico *precision* \times *recall*, possui a métrica *recall* em comum com a curva ROC. Uma representação da matriz de confusão que facilita esta métrica em conjunto é dada na Figura B.1.

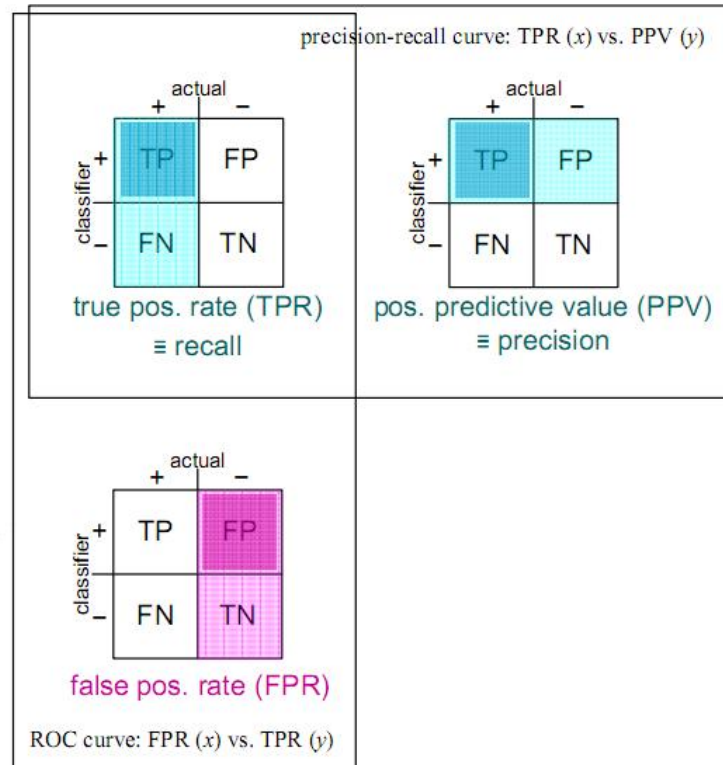


Figura B.1: A matriz de confusão e as métricas que compõe as curvas ROC e *precision* \times *recall*. Fonte: Pres (2008).

Há uma certa discussão sobre qual destes espaços é o melhor. Davis e Goadrich (2006) mostram que ambos possuem a mesma informação, e é possível transformar uma curva no espaço ROC numa curva no espaço *Precision* \times *Recall* e vice-versa. Além disso, mostram também que a curva que é dominante em um espaço, será dominante no outro espaço. Transcreve-se, traduzidos, dois teoremas provados no artigo:

Teorema. Para um dado conjunto de instâncias de valores positivos e negativos, existe uma correspondência de um-para-um entre a curva ROC e a curva *Precision* \times *Recall*, tal que as curvas contém exatamente a mesma matriz de confusão, se $Recall \neq 0$.

Teorema. Para um número fixo de valores positivos e negativos, uma curva domina uma segunda curva no espaço ROC se e apenas se esta curva domina a segunda curva no espaço *Precision* \times *Recall*.

Embora exista uma relação profunda entre ambos os espaços, ROC e $Precision \times Recall$, com correspondência unívoca, a curva ROC possui uma desvantagem em sua apresentação de dados quando há um grande desbalanceamento na distribuição de classes (Drummond e Holte, 2004). Assim, quando o conjunto de dados é muito assimétrico quanto à distribuição das classes, sugere-se o uso de $Precision \times Recall$ (Davis e Goadrich, 2006; Fawcett, 2006), pois suas curvas informam de forma mais adequada o desempenho dos algoritmos. Na Figura B.2, mostra-se o exemplo de Davis e Goadrich (2006), onde temos os mesmos pontos de dois algoritmos sendo comparados nos espaços ROC e $Precision \times Recall$. O ponto ideal numa curva ROC é o ponto (0,1) e na curva $Precision \times Recall$ é o ponto (1,1). A impressão errônea dada pelo gráfico no espaço ROC é que as curvas estão bem próximas do ponto ótimo, enquanto o espaço $Precision \times Recall$ mostra que se pode melhorar a performance dos algoritmos.

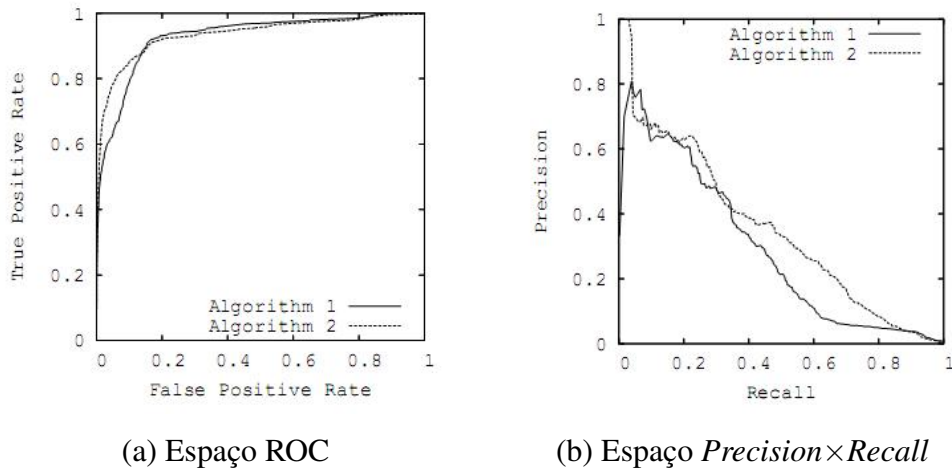


Figura B.2: Dois algoritmos sendo comparados nos espaços ROC e $Precision \times Recall$.
Fonte: Davis e Goadrich (2006)

Na Figura B.3, mostra-se o exemplo de Fawcett (2006), que compara dois algoritmos. Nos gráficos (a) e (b), a distribuição das classes foi balanceada (1:1). Os gráficos (c) e (d) mostram os mesmos dois classificadores no mesmo domínio, porém a distribuição de classes mudou para (1:10). Observe que o gráfico (c) ficou idêntico, enquanto o gráfico (d) mudou substancialmente.

No caso da aplicação de segmentação de imagens, o total P são todos os pixels marcados como borda pelos anotadores humanos e N são os pixels restantes da imagem. Há um grande desbalanceamento da distribuição destas duas classes, pois a quantidade de pixels de borda é bem menor que a quantidade de pixels não-borda.

A Figura B.4 mostra visualmente os conjuntos que são usados para o cálculo das duas métricas: $Precision$ e Recall. O conjunto formado pela elipse à esquerda é o conjunto dos

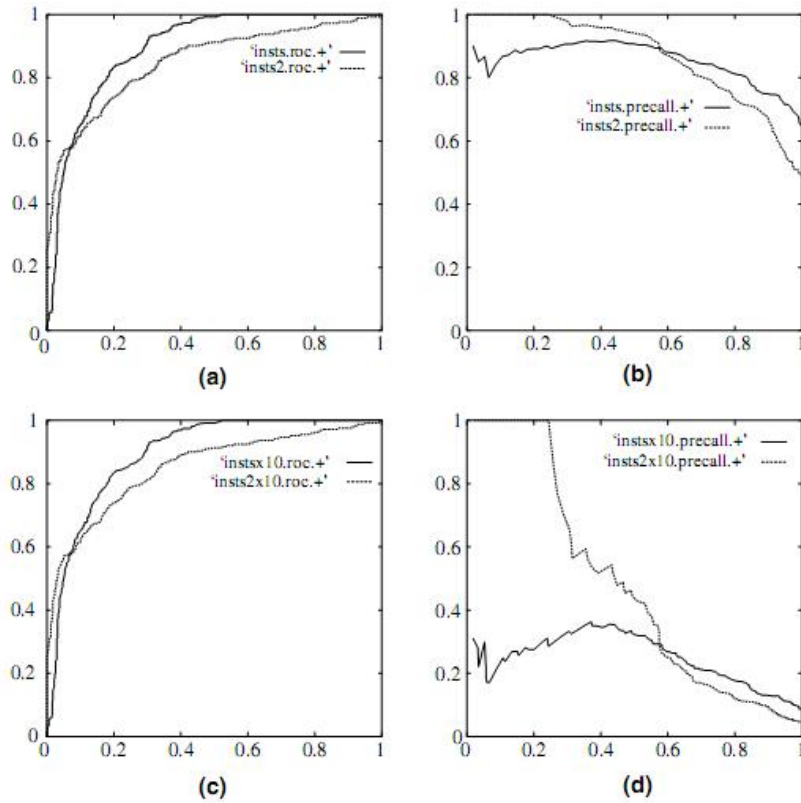


Figura B.3: Curvas ROC e $Precision \times Recall$ em diferentes distribuições de classes (a) curvas ROC 1:1 (b) curvas $Precision \times Recall$ 1:1 (c) curvas ROC 1:10 (d) curvas $Precision \times Recall$ 1:10 . Fonte: Fawcett (2006).

pixels considerados bordas “verdadeiras”, isto é, são os pixels que os anotadores humanos indicaram como borda. O conjunto formado pela elipse com padrão quadriculado (à direita) representa o conjunto dos pixels que o método de segmentação indicou como pixels de borda. A interseção destas duas elipses, o conjunto A, representa os pixels que o método acertou. O conjunto C representa todos os pixels que o método indicou como bordas, mas que não foram indicados como borda pelos anotadores humanos. O conjunto B indica os pixels de borda que os anotadores humanos indicaram como borda, mas que o método não indicou como tal. Assim, as métricas são definidas como:

$$recall = \frac{A}{A+B}, \quad (B.5)$$

$$precision = \frac{A}{A+C}, \quad (B.6)$$

A métrica *precision* é a fração entre a quantidade de pixels de borda detectados corretamente pelo algoritmo sobre a quantidade total de pixels de borda que o próprio algoritmo

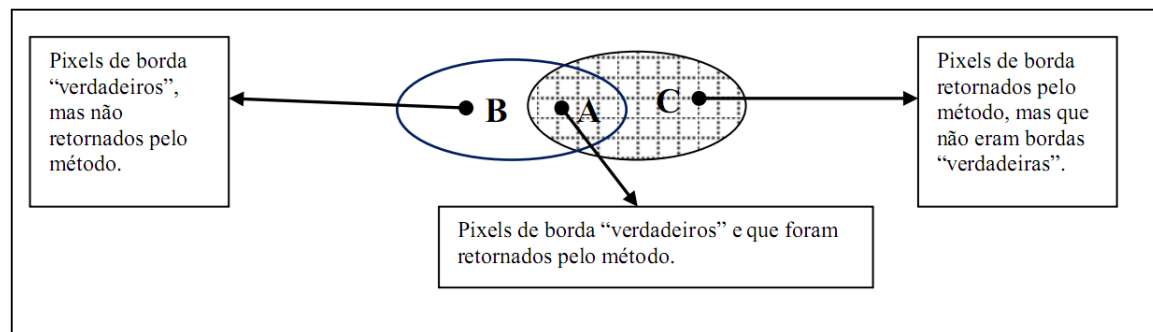


Figura B.4: Conjuntos dos pixels retornados pelo método, que podem ser relevantes ou irrelevantes.

rotulou como borda. *Precision* pode ser visto como uma medida de exatidão ou fidelidade, e *recall* é uma medida de completude. *Recall* é a fração entre a quantidade de pixels de borda que o algoritmo detectou corretamente sobre a quantidade de pixels de borda do *benchmark*. Caso o método indicasse todos os pixels de uma imagem como pixels de borda, o valor de *recall* seria de 1, porém, o valor de *precision* baixo. No entanto, se o método indica apenas um único pixel de borda correto, o valor de *precision* será de 1, mas o valor de *recall* será extremamente baixo. Assim, é comum que essas medidas sejam inversamente relacionadas.

E mais, de acordo com Estrada e Jepson (2009), *precision* e *recall* são particularmente atraentes como medidas de qualidade de segmentação, porque elas não são tendenciosos em favor de sobre ou sub-segmentação de imagens. *Recall* é sensível à sobre-segmentação em contrapartida *precision* é sensível à sub-segmentação.

Apêndice C

BSDS

Na introdução foi discutida a ambiguidade intrínseca da percepção visual humana. Para resolver uma parte desta questão, utilizou-se o banco de dados “*The Berkeley Segmentation Dataset and Benchmark*” (BSDS) (Martin et al., 2001), pois este banco fornece um referencial humano coerente e métricas-padrão que permitem uma comparação quantitativa entre a performance de diferentes algoritmos e com o resultado humano.

No BSDS há duas bases de dados contendo imagens coloridas: uma base de treinamento, composta por 200 imagens, e uma base de testes, composta por 100 imagens. Todas as imagens possuem tamanho de 481×321 pixels. A base de treinamento foi proposta para calibração ou para a fase de aprendizado de algoritmos, e não faz parte do cômputo da métrica *F-measure* de avaliação de métodos. Como não há fase de aprendizado nem calibração nesta Tese, usou-se a base de dados de treino apenas para testes qualitativos, mas não para os testes quantitativos. Para cada imagem da base de testes há pelo menos cinco segmentações manuais feitas por anotadores humanos diferentes, compondo, assim, uma referência de comparação.

Foi um trabalho que durou oito meses apenas para a aquisição das anotações humanas, somando um total de 1.458 horas de trabalho de 30 anotadores. Além desses meses, houve tempo de desenvolvimento da ferramenta de segmentação para o anotador humano, tempo de seleção dos anotadores e desenvolvimento das ferramentas de cálculos de métricas.

O sistema de anotação humana escolhia aleatoriamente uma imagem para cada anotador. Para cada imagem, o sistema também escolhia a forma da apresentação da imagem. Conforme Figura C.1, a imagem poderia ser apresentada (a) em sua forma normal, (b) em escala de cinza ou (c) invertida-negativa. Com isso, esperava-se diminuir o impacto que o conhecimento prévio gera na segmentação do anotador humano.



Figura C.1: Exemplo das diferentes formas de apresentação para o anotador humano. Fonte: Martin (2002)

C.1 A Consistência das Anotações Humanas

Após a produção de 11.559 anotações humanas de 1.020 imagens, a questão que surge é: elas são consistentes entre elas?

Para responder à esta questão, o trabalho de construção do BSDS usou a teoria da árvore de percepção de Palmer (Palmer, 1999) como modelo de organização perceptiva de uma cena. A Figura C.2 mostra em (a) a imagem de entrada à esquerda e sua árvore de percepção associada à direita. Em (b), (c) e (d) mostra-se três diferentes anotações humanas à esquerda e as suas correspondentes árvores de percepção à direita. A consistência é verificada pela estrutura da árvore ter-se mantido, com a supressão de algumas ramificações, mas não houve mudança em sua semântica.

Na Figura C.3 repete-se a imagem da sereia e ao lado, em sequência, mostra-se duas anotações humanas. A segunda anotação humana é inconsistente com a primeira anotação humana. Neste caso, as duas segmentações não são consistentes entre si porque não há nenhuma única decomposição hierárquica da cena que pode explicar ambas ao mesmo tempo. A inconsistência se manifesta pelo fato do segundo anotador ter segmentado uma área de sombra entre as duas pedras, mas não indicando uma fronteira entre as duas pedras. Além disso, esta área de sombra não representa um elemento semântico na cena.

O trabalho do BSDS analisou a consistência entre as várias anotações humanas e selecionou as imagens em que todas as anotações humanas se adequassem ao critério da árvore de percepção.

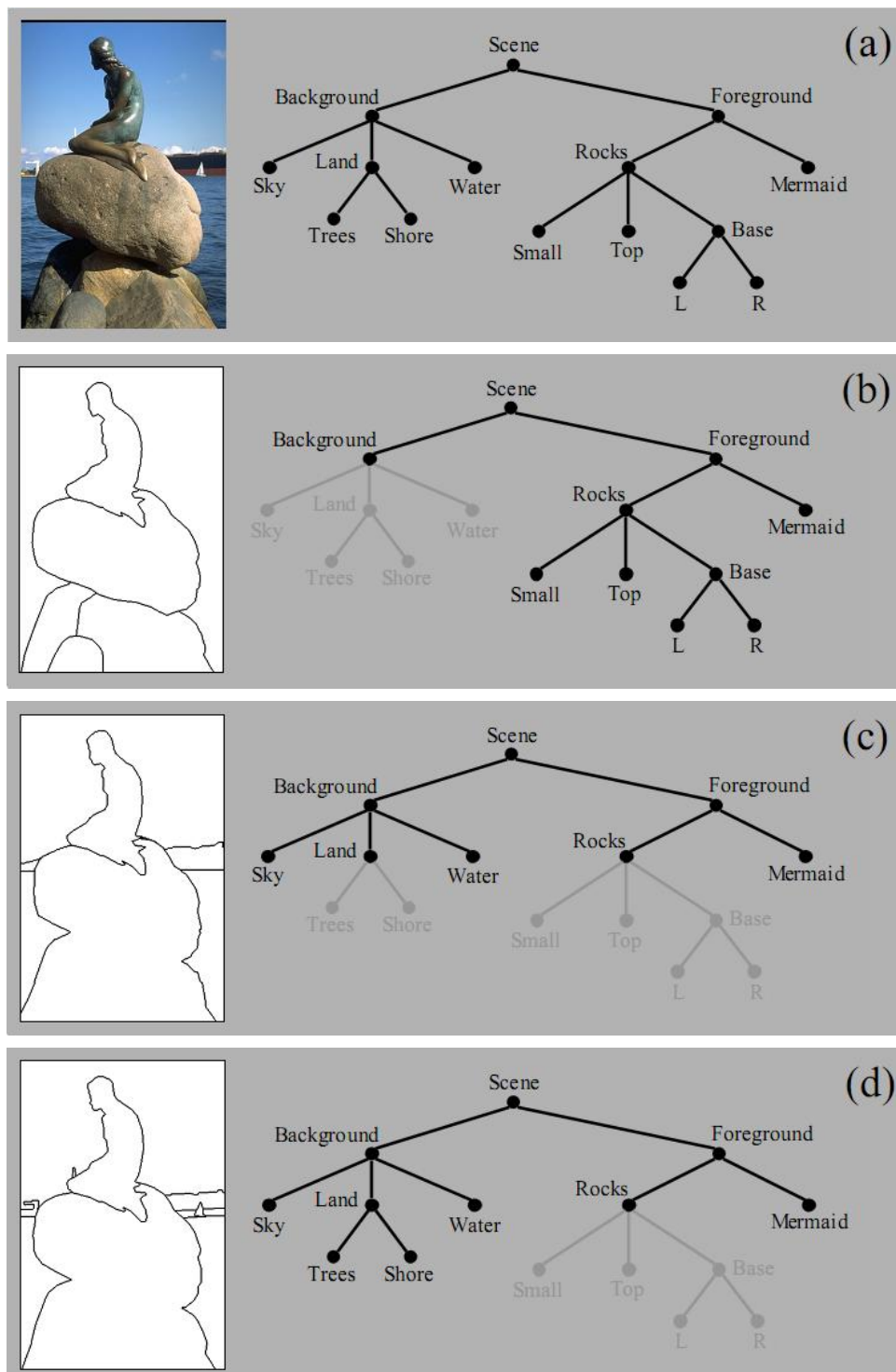


Figura C.2: Exemplos de segmentações a sua árvore de percepção associada. Fonte: Martin (2002)

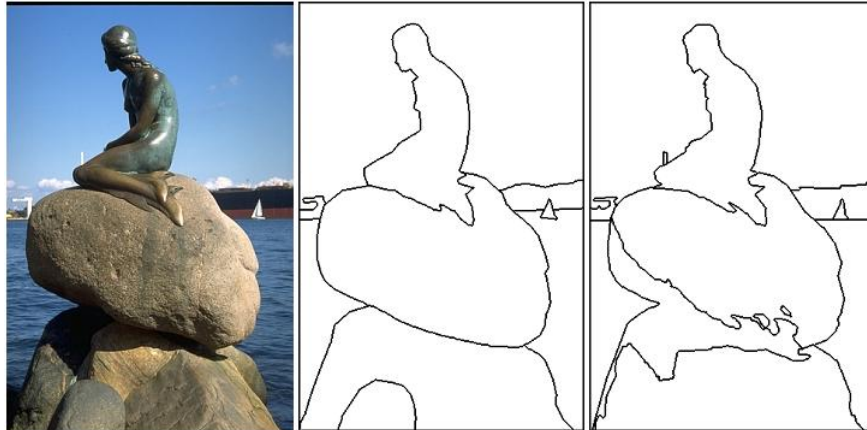


Figura C.3: Exemplo de inconsistência entre anotações humanas. Fonte: Martin (2002)

C.2 O Cálculo das Métricas

Nesta seção, apresentamos o algoritmo usado para computar a correspondência entre um mapa de fronteiras binarizado gerado por um sistema computacional e um mapa de fronteiras humano. O problema é convertido para um problema de custo mínimo de um grafo bipartido, onde o peso entre um pixel do mapa gerado por um sistema e um pixel anotado por um ser humano é proporcional à sua distância relativa no plano de imagem. O grafo de menor custo é aquele que tiver a menor distância entre os dois mapas, isto é, a soma das distâncias entre dois pontos correspondentes é a menor.

No campo da matemática da teoria dos grafos, um grafo bipartido ou bigrafo é um grafo cujos vértices podem ser divididos em dois conjuntos disjuntos U e V tais que toda aresta conecta um vértice em U a um vértice em V ; ou seja, U e V são conjuntos independentes, veja um grafo bipartido simples na Figura C.4.

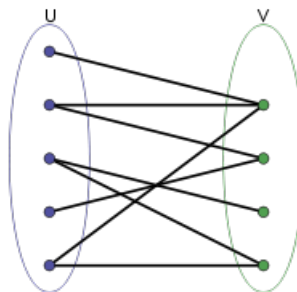


Figura C.4: Exemplo de um grafo bipartido simples.

De uma forma simplificada, a comparação entre duas segmentações S_1 e S_2 é feita como considerando S_1 sendo o conjunto U e S_2 sendo o conjunto V . A Figura C.5 mostra na sua

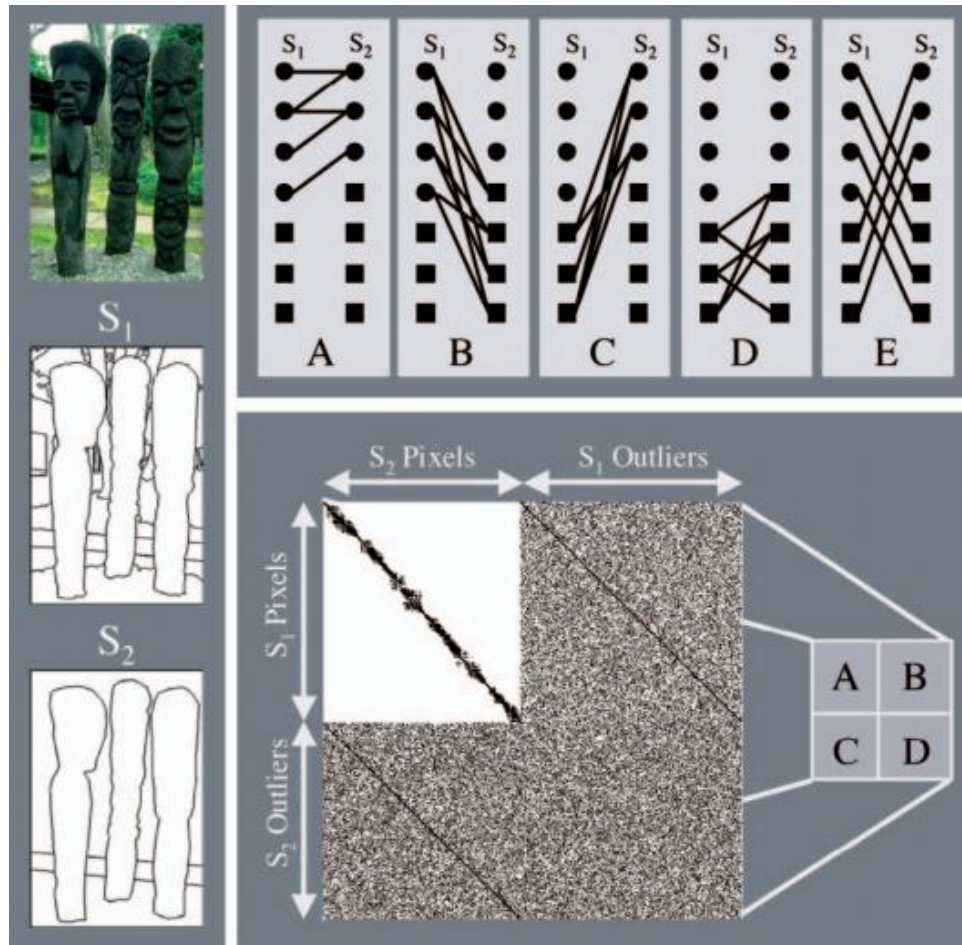


Figura C.5: Esquema simplificado dos passos para a comparação entre duas segmentações.
Fonte: Martin et al. (2004)

parte esquerda, a imagem original e duas segmentações a serem comparadas. Na parte superior da Figura C.5, temos as possíveis combinações de emparelhamento entre os conjuntos U e V . Sendo que é definida uma tolerância máxima entre o emparelhamento, isto é, há um limite de distância entre um elemento do conjunto U e um elemento do conjunto V . A escolha é feita pela representação de menor custo.

Na parte inferior da Figura C.5, mostra-se uma representação gráfica da matriz de adjacências. No exemplo da figura, S_1 possui $n_1=3.664$ nós e S_2 possui $n_2=4.124$ nós. A matriz de adjacências é uma matriz $n \times n$, onde $n = n_1 + n_2=7.788$. A matriz de adjacências de um grafo bipartido pode ser compreendido como um conjunto de quatro partes: A, B, C e D. A teoria de grafos indica que apenas as diagonais das partes B e C são os elementos que representam unicamente o grafo. Para uma dada segmentação de uma imagem do BSDS, este cômputo é feito para cada anotação humana. A matriz de confusão acumula o resultado de todas as comparações com a anotação humana, e com isso, gera apenas uma única métrica de *recall* e *precision*.

Mesmo as imagens das anotações humanas são comparadas desta forma, isto é, se há 5 anotações diferentes, é feita uma comparação par a par entre elas. É por isso que mesmo a anotação humana não possui um valor de *F-measure* igual ao máximo (1). Quanto menor é o valor *F-measure*, mais diferentes foram as anotações humanas entre elas. De outra forma, quanto menor o valor de *F-measure*, significa que houve maior discordância entre o entendimento dos anotadores humanos.

Quando a imagem de entrada é uma imagem em escala de cinza, o BSDS tem um parâmetro que indica a quantidade de diferentes limiares a serem aplicados para a binarização da imagem. Para cada limiar, então calcula-se as métricas de *recall*, *precision* e *F-measure*, gerando uma curva no espaço *Precision-Recall*. O valor de *F-measure* desta imagem é referente ao ponto mais próximo do objetivo (1,1).

Para determinar o *F-measure* de um algoritmo, existe uma única matriz de confusão que acumula o resultado das 100 imagens da base de testes. Gerando uma métrica única, que corresponde ao seu resumo estatístico. Quanto maior o *F-measure* do algoritmo, mais seu resultado se aproxima do *benchmark* humano.

Apêndice D

Multifractal

O termo fractal foi cunhado por Mandelbrot (1983) para descrever a estrutura irregular de muitos objetos e fenômenos naturais. Em termos bem simples, um fractal é um objeto composto por partes que são auto-similares e invariantes à escala.

Desde que Pentland (1984) apresentou evidências de que a maioria das superfícies naturais são fractais, e que a superfície de imagens digitais de cenas naturais também são fractais, a teoria de fractais foi aplicada com sucesso em várias áreas de processamento de imagens (Pesquet-Popescu e Lévy Véhel, 2002). Em princípio, um fractal criado matematicamente é auto-similar sobre uma faixa infinita de escalas, enquanto imagens digitais fractais têm uma faixa limitada de escalas (Pentland, 1984).

A base da teoria fractal está no conceito de auto-similaridade que, no espaço Euclidiano, pode ser formulado da seguinte maneira (Turner et al., 1998):

Conceito 1. *Considere um conjunto limitado F em um n -espaço Euclidiano. O conjunto F é dito ser auto-similar se F está na união de K distintas (não-sobrepostas) cópias de si mesmo, em que cada cópia está escalada para baixo por uma razão menor que um em todas as coordenadas.*

O conceito de auto-similaridade no espaço Euclidiano foi estendido, a auto-similaridade pode ser euclidiana ou estatística. Isto significa que a forma é similar ao padrão de uma pequena parte arbitrária de si mesma ou pode ser uma auto-similaridade estatística, isto é, suas medidas estatísticas são invariantes à escala.

1. *Objeto Auto-Similar Determinístico:* no qual o fractal (então denominado fractal Euclidiano) é composto de características distintas que remontam cada outro objeto idêntico em diferentes escalas.

2. *Objeto Auto-Similar Estatístico*: no qual as características do fractal podem mudar em diferentes escalas, mas suas propriedades estatísticas em todas as escalas são as mesmas.

A Figura D.1a mostra um exemplo de um fractal determinístico, a parte (b) mostra a parte retangular da parte (a) ampliada 6 vezes e a parte (c) mostra a parte retangular da parte (b) ampliada 100 vezes. A Figura D.2 mostra dois exemplos de fractais estatísticos.

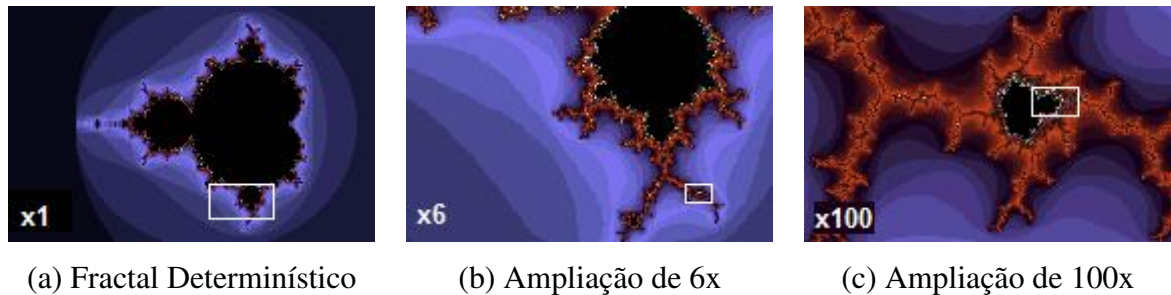


Figura D.1: Fractal de Mandelbrot.

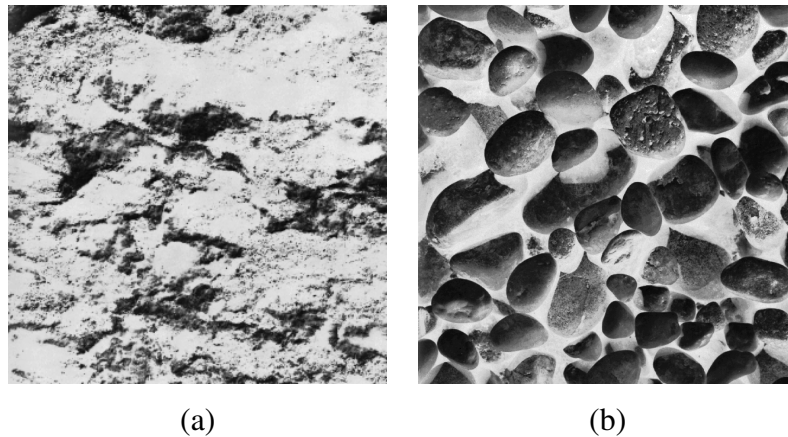


Figura D.2: Fractais estatísticos.

A dimensão fractal é uma medida que reflete o grau de irregularidade sobre múltiplas escalas. É um número não inteiro e, numa imagem, caracteriza a percepção humana de rugosidade. Segundo Kasparis e colegas (Kasparis et al., 2001), existem vantagens em usar a dimensão fractal sobre outras características de imagens, pois esta medida é insensível a transformações lineares nos níveis de cinza da imagem e ao ruído multiplicativo.

No entanto, a dimensão fractal é uma medida estatística da imagem como um todo. Ela representa uma medida global, e quantifica as propriedades da imagem como se fosse um único fractal. Mas, uma cena natural pode conter informações diferentes em sua extensão, e daí a dimensão fractal pode variar conforme a localização (Revathy, 2007). A estrutura

pode ser considerada como uma mistura de diferentes fractais, cada um com dimensão diferente. Em tal situação, é necessário se definir um espectro que dê a informação de como é a distribuição das dimensões fractais da estrutura. Este é o conceito de multifractal. O termo foi inicialmente proposto por Frisch (1980), e depois este conceito foi utilizado em vários outros artigos em diversas áreas (Zharkova e Jain, 2007; Lam, 1998).

Conforme Chaudhuri e Sarkar (1995), a dimensão mono-fractal caracteriza a auto-similaridade em casos ideais, e muitas texturas reais não são fractais ideais. No trabalho de Peleg e colegas (Peleg et al., 1984), é mostrado que texturas reais são muitas vezes semi-fractais e anisotrópicos. Um semi-fractal (Creutzburg e Ivanov, 1989) foi um termo cunhado para referenciar sistemas que possuam auto-similaridade, mas esta similaridade não é exata. O conceito de multifractal é bem mais genérico, conforme Stanley e Meakin (1988), pois não é necessário se ter uma estrutura fractal para se encontrar um fenômeno multifractal. Pesquet-Popescu e Lévy Véhel (2002) mostram que a análise multifractal modela bem casos determinísticos e aleatórios, fenômenos anisotrópicos e não-estacionários. Multifractais também foram usados com sucesso na detecção de fronteiras entre regiões (Revathy, 2007; Côco et al., 2007). Enfim, a teoria multifractal abrange diversos fenômenos caracterizados por correlações espaciais de uma forma quantitativa.

Multifractais podem ser vistos como uma extensão de fractais (Lopes e Betrouni, 2009). O sistema fractal é caracterizado por uma lei de escala, com um expoente fractal, isto é, um expoente não inteiro. Um sistema multifractal é uma generalização de um sistema fractal em que um único expoente (dimensão fractal) não é bastante para descrever sua dinâmica; em vez disso, um conjunto de expoentes é necessário.

Sistemas multifractais são comuns na natureza, especialmente geofísica. Eles incluem a turbulência, o estoque de séries temporais do mercado, as cenas do mundo real, a dinâmica do campo magnético do Sol e a dinâmica de batimentos cardíacos. Em um sistema de multifractal, o comportamento ao redor de qualquer ponto é descrito por uma lei de potência local (Lashermes et al., 2008):

$$|s(x+a) - s(x)| \sim a^{q(x)}, \text{ onde } |a| \rightarrow 0. \quad (\text{D.1})$$

O expoente $q(x)$ é chamado de expoente de singularidade, pois ele descreve o grau de singularidade local ou regularidade em torno do ponto x .

O conjunto formado por todos os pontos que compartilham o mesmo expoente de singularidade é um conjunto fractal de dimensão multifractal D_q . A curva D_q versus q é chamado de espectro de singularidade e descreve completamente a distribuição (estatística) da variável s .

Diferentes técnicas foram propostas para avaliar a dimensão fractal, avaliando o quão rápido é o crescimento de seu comprimento, área ou volume, com relação às escalas cada vez menores. Existem várias abordagens (Lopes e Betrouni, 2009): *Boxcounting methods*, *Variogram methods*, *Triangular Prism method*, *Epsilon-Blanket method*, *Isarithm method* e *Power Spectrum method*. Para o cálculo da dimensão multifractal (Lopes e Betrouni, 2009), há basicamente duas classes de métodos: extensões do método de *Box-Counting* e os métodos baseados em *wavelets*.

D.1 O Método *Differential Box-Counting*

Box-Counting é o método mais popular para o cálculo de dimensão multifractal, e foi originalmente desenvolvido por Voss (1986). Uma versão muito eficiente é a versão de Sarkar e Chaudhuri (1994), chamada *Differential Box-Counting* (DBC). Neste trabalho, se faz a estimativa da dimensão fractal da imagem como um todo. Um ano depois, os mesmos autores apresentaram um artigo no qual eles mostram como usar o DBC para o cálculo da dimensão multifractal (Chaudhuri e Sarkar, 1995; Sarkar e Chaudhuri, 1995). Nestes trabalhos, define-se uma estimativa de forma local, isto é, divide-se a imagem em janelas locais, assim como nos trabalhos de (Véhel e Mignot, 1994; Maeda et al., 2000; Kasparis et al., 2001). No DBC, a medida é obtida pela contagem da menor quantidade de cubos de um determinado tamanho que podem cobrir inteiramente toda a superfície. A cada alteração do tamanho do cubo, refaz-se a contagem. O método DBC é uma aproximação do valor ótimo, mas é de simples implementação e computacionalmente eficiente.

O processo deve ser feito visualizando a imagem como uma superfície 3D, onde (i, j) denota a posição 2D e a terceira coordenada, z , é o valor em escala de cinza do pixel em questão. Considere-se uma imagem de tamanho $M \times M$. Há um cubo deslizante de aresta r , um número inteiro tal que $1 < r \leq M/2$. Dependendo dos valores dos pixels, será necessário empilhar cubos, fazendo uma coluna de cubos (Figura D.3).

Digamos que o máximo e mínimo dos valores de níveis de cinza que se encontram na posição (i, j) recaem no v -ésimo e no u -ésimo cubo, respectivamente. Então o número de cubos, η_r , é calculado como

$$\eta_r(i, j) = v - u + 1, \quad (\text{D.2})$$

Pode-se imaginar como se existisse uma superfície de máximo e uma outra superfície de mínimo e que η_r é a diferença entre estas superfícies ponto a ponto. O número total de cubos

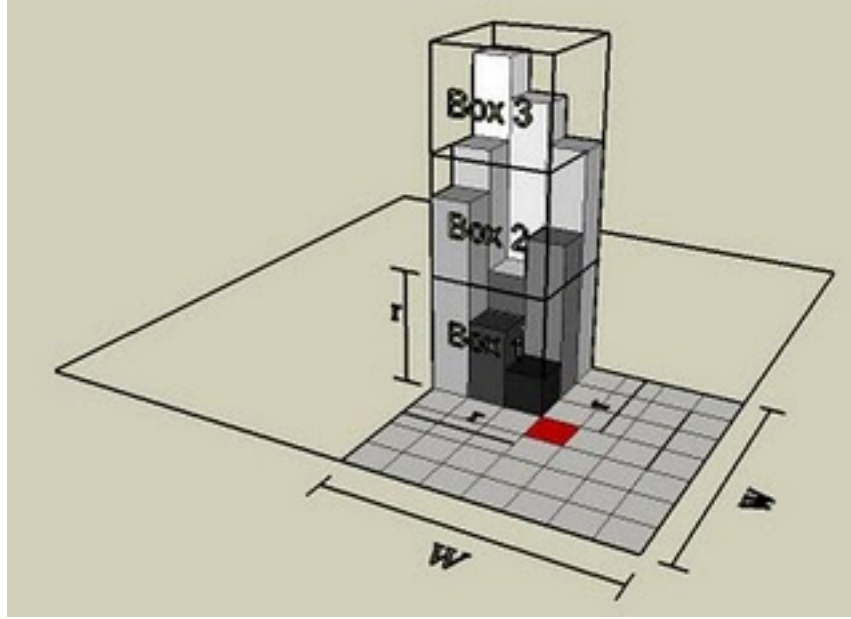


Figura D.3: Exemplo do algoritmo DBC. Fonte: Barros Filho e Sobreira (2008).

(ou a massa da imagem), N_r , necessário para cobrir toda a imagem pode então ser estimado como

$$N_r = \sum_{i,j} \eta_r(i, j). \quad (\text{D.3})$$

A medida de distribuição de diferentes subfractais em cada ponto, $\mu_r(i, j)$, é definida como

$$\mu_r(i, j) = \frac{\eta_r(i, j)}{N_r}, \quad (\text{D.4})$$

onde η_r e N_r são dados pelas equações D.2 e D.3, respectivamente. Já a função de partição é dada por

$$\chi_q = \sum_{i,j} \mu_r(i, j)^q. \quad (\text{D.5})$$

O somatório dos valores μ_r se refere à função de partição. O objetivo desta função é avaliar todas as possíveis singularidades da distribuição, estudando todos os momentos de ordem q . Chega-se à dimensão multifractal de ordem q , que é estimada como

$$(q-1)D_q \equiv \tau(q) = \lim_{r \rightarrow 0} \frac{\ln [\sum_{i,j} \mu_r(i, j)^q]}{\ln \left(\frac{M}{r} \right)} = \lim_{r \rightarrow 0} \frac{\ln(\chi_q)}{\ln \left(\frac{M}{r} \right)}, \quad (\text{D.6})$$

onde $q \neq 1$ e representa a ordem estatística de auto-similaridade. Para um sistema monofractal, a equação D.6 apresenta sempre o mesmo valor, independente do valor de q .

Singularidades são formas do tipo degrau (*step-like*) ou cúspide (*cusp-like*), (Chávez et al., 2010; Ivanov et al., 1999; Plamen et al., 2001). A Figura D.4 mostradas estas singularidades em um gráfico 2D, de mais fácil visualização.

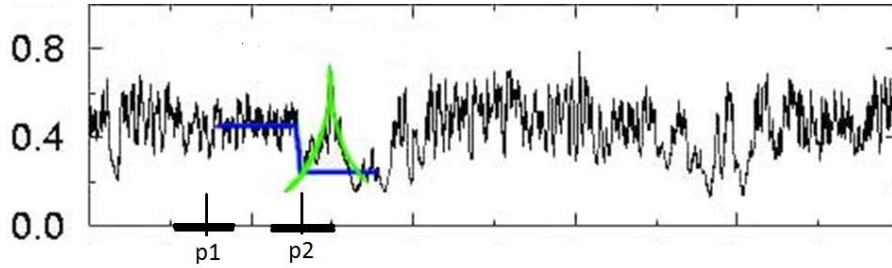


Figura D.4: Singularidades do tipo degrau e cúspide.

Para entender como o resultado diferencia as singularidades, considere dois pontos: $p1$ e $p2$ da Figura D.4. O ponto $p1$ e seu intervalo abrangem a parte do sinal que possui uma variação em torno de seu valor médio que expressa uma rugosidade regular e o ponto $p2$ se encontra exatamente no ponto de mudança de descontinuidade do degrau. Considerando um intervalo de mesmo comprimento para cada ponto, avalia-se comparativamente os valores das equações para cada ponto. O resultado da equação D.2 tem um maior para o ponto $p2$ que para o ponto $p1$, pois é maior a diferença entre o maior valor e o menor valor do eixo y no intervalo x , consequentemente a equação D.3 também apresenta maior valor em $p2$ do que em $p1$. O resultado da equação D.3 é o divisor na equação D.4, assim, a medida de distribuição e a função de partição D.5 são menores no ponto $p2$ do que no ponto $p1$.

Significa que para os pontos de singularidades, a função de partição χ_q possui valor menor. Logo, os degraus e os cúspides terão valores mais escuros numa imagem. Veja um exemplo na Figura D.5, onde em (a) temos a imagem de entrada e em (b) a imagem resultante da função de partição local. Nos pontos onde há alteração de rugosidade e iluminação, isto é, as divisões entre texturas, os pixels da imagem do descritor fractal local são mais escuros.

A imagem de entrada pode ser particionada em janelas locais de tamanho $W \times W$. Assim, a equação D.3 será calculada sobre todos os elementos da janela local de tamanho W , obtendo-se a massa daquela janela local. Com este valor, a distribuição μ_r (equação D.4) também será uma informação sobre a janela. Na equação D.6, troca-se M por W , e se obtém a dimensão da janela local.



(a) Imagem de Entrada

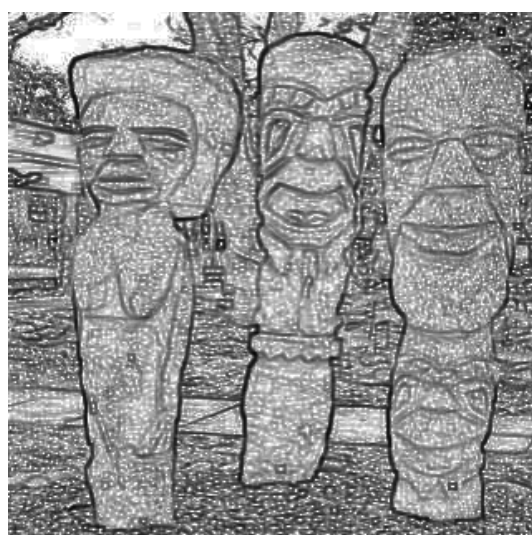
(b) $q = 2, W = 3$

Figura D.5: Imagem do descritor multifractal.